Oracle® Retail Store Inventory Management

Installation Guide
Release 13.2.1
E22217-05

June 2013

ORACLE

Oracle® Retail Store Inventory Management Installation Guide, Release 13.2.1

Copyright © 2011, Oracle. All rights reserved.

Primary Author: Donna Linde

Contributors: Nathan Young

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Value-Added Reseller (VAR) Language

Oracle Retail VAR Applications

The following restrictions and provisions only apply to the programs referred to in this section and licensed to you. You acknowledge that the programs may contain third party software (VAR applications) licensed to Oracle. Depending upon your product and its version number, the VAR applications may include:

(i) the **MicroStrategy** Components developed and licensed by MicroStrategy Services Corporation (MicroStrategy) of McLean, Virginia to Oracle and imbedded in the MicroStrategy for Oracle Retail Data Warehouse and MicroStrategy for Oracle Retail Planning & Optimization applications.

(ii) the **Wavelink** component developed and licensed by Wavelink Corporation (Wavelink) of Kirkland, Washington, to Oracle and imbedded in Oracle Retail Mobile Store Inventory Management.

(iii) the software component known as **Access Via™** licensed by Access Via of Seattle, Washington, and imbedded in Oracle Retail Signs and Oracle Retail Labels and Tags.

(iv) the software component known as **Adobe Flex™** licensed by Adobe Systems Incorporated of San Jose, California, and imbedded in Oracle Retail Promotion Planning & Optimization application.

You acknowledge and confirm that Oracle grants you use of only the object code of the VAR Applications. Oracle will not deliver source code to the VAR Applications to you. Notwithstanding any other term or condition of the agreement and this ordering document, you shall not cause or permit alteration of any VAR Applications. For purposes of this section, "alteration" refers to all alterations, translations, upgrades, enhancements, customizations or modifications of all or any portion of the VAR Applications including all reconfigurations, reassembly or reverse assembly, re-engineering or reverse engineering and recompilations or reverse compilations of the VAR Applications or any derivatives of the VAR Applications. You acknowledge that it shall be a breach of the agreement to utilize the relationship, and/or confidential information of the VAR Applications for purposes of competitive discovery.

The VAR Applications contain trade secrets of Oracle and Oracle's licensors and Customer shall not attempt, cause, or permit the alteration, decompilation, reverse engineering, disassembly or other reduction of the VAR Applications to a human perceivable form. Oracle reserves the right to replace, with functional equivalent software, any of the VAR Applications in future releases of the applicable program.

Contents

Send Us Your Comments ..... **ix**

Preface ..... **xi**

Audience xi
Related Documents xi
Customer Support xi
Review Patch Documentation xi
Oracle Retail Documentation on the Oracle Technology Network xii
Conventions xii

1 Preinstallation Tasks **1**

Check for the Current Version of the Installation Guide 1
Implementation Capacity Planning 1
Upgrading SIM 1
Check Supported Database Server Requirements 2
Check Supported Application Server Requirements 3
Check Single Sign-On Requirements 4
Check Directory Server Requirements 4
Check Third-Party Software Dependencies 4
Check Client PC and Web Browser Requirements 5
Supported Oracle Retail Products 5
UNIX User Account Privileges to Install the Software 6
SIM Installation Overview 6
Customer Profiles 6

2 RAC and Clustering **7**

3 Database Installation Tasks **9**

Expand the SIM Database Schema Installer Distribution 9
Create Tablespaces 9
Create the SIM Database User 10
Run the SIM Database Schema Installer 10
Resolving Errors Encountered During Database Schema Installation 11

4 Application Installation **13**

Application Server Deployment Options 13

5 Installing the SIM Application on Oracle Application Server (OAS) **15**

Oracle Configuration Manager 15
Create a New OC4J Instance and Group for SIM 15
Configure Apache for JNLP Files 16
Set the LANG Environment Variable 17
Expand the SIM Application Distribution 17
Run the SIM Application Installer 17
Review and/or Configure Oracle Single Sign-On 18

SIM Batch Scripts 19
Adding Users to Application Server for Web Services 19
Resolving Errors Encountered During Application Installation 22
Backups Created by Installer 22
Web Help Files 22
Starting and Stopping SIM 22
Starting and Stopping the Wavelink Server 22

6 Installing the SIM Application on WebLogic 25
Oracle Configuration Manager 25
Install Managed Server in WebLogic 25
Install NodeManager 29
Start the Managed Servers 36
Set the LANG Environment Variable 39
Expand the SIM Application Distribution 39
Run the SIM Application Installer 39
Clustered Installations – Post-Installation Steps 40
Review and/or Configure Oracle Single Sign-On 40
SIM Batch Scripts 41
Configuring Web Service Security in SIM 41
Resolving Errors Encountered During Application Installation 43
Web Help Files 43
Starting and Stopping the Wavelink Server 43

7 Data Seeding 45

8 Test the SIM Application 47

A Appendix: SIM Database Schema Installer Screens 49

B Appendix: SIM Application Oracle Application Server (OAS) Installer Screens. 57

C Appendix: SIM Application WebLogic Server Installer Screens 87

D Appendix: Installer Silent Mode 113
Repeating an Installation Attempt 113

E Appendix: Common Installation Errors 114
EJB Deployment Errors during Installation to WebLogic 114
XML Processing Errors While configuring sim-client.ear or sim-server.ear 114
Output Freezes during Text Mode Installation to OAS and WebLogic 115
Database Installer Hangs on Startup 115
Unreadable Buttons in the Installer 115
Message: Unable to get a deployment manager 116
Warning: Could not create system preferences directory 116
Keystore Errors When Signing sim-config.jar 117
Warning: Couldn't find X Input Context 117
ConcurrentModificationException in Installer GUI 117
Error while unpacking the ear file 117

A Second Login Screen Appears After Single Sign-On Login 118
Error Connecting to Database URL 118
Installer Fails because of missing .jar in $ORACLE_HOME/utils/ccr/lib 119
Files not available to copy at the end of installation results in non working applications – WebLogic only 119

F Appendix: Database Parameter File 121

G Appendix: Oracle Single Sign-On for Oracle Application Server (OAS) 123
What Do I Need for Oracle Single Sign-On? 123
Can Oracle Single Sign-On Work with Other SSO Implementations? 123
Oracle Single Sign-on Terms and Definitions 124
What Single Sign-On is not 125
How Oracle Single Sign-On Works 125
Installation Overview 127
User Management 128
Setting up SIM for Single Sign-on 128

H Appendix: Oracle Single Sign-On for WebLogic 131
What Do I Need for Oracle Single Sign-On? 131
Can Oracle Single Sign-On Work with Other SSO Implementations? 132
Oracle Single Sign-on Terms and Definitions 132
What Single Sign-On is not 133
How Oracle Single Sign-On Works 133
Installation Overview 135
User Management 136
How to Configure Single Sign On Functionality on SIM 137

I Appendix: Setting Up Password Stores with Oracle Wallet 143
About Password Stores and Oracle Wallet 143
Setting Up Password Stores for Database User Accounts 144
Setting Up Wallets for Database User Accounts 145
For RMS, RWMS, RPM Batch, RETL, RMS, RWMS, and ARI 145
For Java Applications (SIM, ReIM, RPM, Alloc, RIB, RSL, AIP, RETL) 147
How does the Wallet tie back to the Application? 149
How does the Wallet tie back to java pgm batch use (such as ReIM batch)? 149
Setting up RETL Wallets for 13.2 149
Quick Guide for Retail Wallets 151

J Appendix: Installation Order 157
Enterprise Installation Order 157

Send Us Your Comments

Oracle Retail Store Inventory Management Installation Guide, Release 13.2.1

Oracle welcomes customers' comments and suggestions on the quality and usefulness of this document.

Your feedback is important, and helps us to best meet your needs as a user of our products. For example:

- Are the implementation steps correct and complete?
- Did you understand the context of the procedures?
- Did you find any errors in the information?
- Does the structure of the information help you with your tasks?
- Do you need different information or graphics? If so, where, and in what format?
- Are the examples correct? Do you need more examples?

If you find any errors or have any other suggestions for improvement, then please tell us your name, the name of the company who has licensed our products, the title and part number of the documentation and the chapter, section, and page number (if available).

> **Note:** Before sending us your comments, you might like to check that you have the latest version of the document and if any concerns are already addressed. To do this, access the new Applications Release Online Documentation CD available on My Oracle Support and www.oracle.com. It contains the most current Documentation Library plus all documents revised or released recently.

Send your comments to us using the electronic mail address: retail-doc_us@oracle.com

Please give your name, address, electronic mail address, and telephone number (optional).

If you need assistance with Oracle software, then please contact your support representative or Oracle Support Services.

If you require training or instruction in using Oracle software, then please contact your Oracle local office and inquire about our Oracle University offerings. A list of Oracle offices is available on our Web site at www.oracle.com.

Preface

Oracle Retail Installation Guides contain the requirements and procedures that are necessary for the retailer to install Oracle Retail products.

Audience

This Installation Guide is written for the following audiences:

- Database administrators (DBA)
- System analysts and designers
- Integrators and implementation staff

Related Documents

For more information, see the following documents in the Oracle Retail Store Inventory Management Release 13.2.1 documentation set:

- *Oracle Retail Store Inventory Licensing Information*
- *Oracle Retail Store Inventory Management Data Model*
- *Oracle Retail Store Inventory Management Implementation Guide*
- *Oracle Retail Store Inventory Management Online Help*
- *Oracle Retail Store Inventory Management Operations Guide*
- *Oracle Retail Store Inventory Management Release Notes*
- *Oracle Retail Store Inventory Management User Guide*

Customer Support

To contact Oracle Customer Support, access My Oracle Support at the following URL:

https://support.oracle.com

When contacting Customer Support, please provide the following:

- Product version and program/module name
- Functional and technical description of the problem (include business impact)
- Detailed step-by-step instructions to re-create
- Exact error message received
- Screen shots of each step you take

Review Patch Documentation

When you install the application for the first time, you install either a base release (for example, 13.2) or a later patch release (for example, 13.2.1). If you are installing the base release and additional patch and bundled hot fix releases, read the documentation for all releases that have occurred since the base release before you begin installation. Documentation for patch and bundled hot fix releases can contain critical information related to the base release, as well as information about code changes since the base release.

Oracle Retail Documentation on the Oracle Technology Network

Documentation is packaged with each Oracle Retail product release. Oracle Retail product documentation is also available on the following Web site:
http://www.oracle.com/technology/documentation/oracle_retail.html

(Data Model documents are not available through Oracle Technology Network. These documents are packaged with released code, or you can obtain them through My Oracle Support.)

Documentation should be available on this Web site within a month after a product release.

Conventions

Navigate: This is a navigate statement. It tells you how to get to the start of the procedure and ends with a screen shot of the starting point and the statement, "the Window Name window opens."

```
This is a code sample
    It is used to display examples of code
```

1

Preinstallation Tasks

This chapter discusses the tasks to complete before installation.

Check for the Current Version of the Installation Guide

Corrected versions of Oracle Retail installation guides may be published whenever critical corrections are required. For critical corrections, the rerelease of an installation guide may not be attached to a release; the document will simply be replaced on the Oracle Technology Network Web site.

Before you begin installation, check to be sure that you have the most recent version of this installation guide. Oracle Retail installation guides are available on the Oracle Technology Network at the following URL:

http://www.oracle.com/technology/documentation/oracle_retail.html

An updated version of an installation guide is indicated by part number, as well as print date (month and year). An updated version uses the same part number, with a higher-numbered suffix. For example, part number E123456-**02** is an updated version of an installation guide with part number E123456-**01**.

If a more recent version of this installation guide is available, that version supersedes all previous versions. Only use the newest version for your installation.

Implementation Capacity Planning

There is significant complexity involved in the deployment of Oracle Retail applications, and capacity planning is site specific. Oracle Retail strongly suggests that before installation or implementation you engage your integrator (such as the Oracle Retail Consulting team) and hardware vendor to request a disk sizing and capacity planning effort.

Sizing estimates are based on a number of factors, including the following:

- Workload and peak concurrent users and batch transactions
- Hardware configuration and parameters
- Data scarcity
- Application features utilized
- Length of time history is retained

Additional considerations during this process include your high availability needs as well as your backup and recovery methods.

Upgrading SIM

By default, SIM 13.2.1 is a full installation. It is possible to upgrade a previous release (for example, from SIM 13.2.0.3) installation to version SIM 13.2.1. If you would like to perform an upgrade from SIM 13.2.0.x, refer to the My Oracle Support document, *Oracle Retail Upgrade Guide* (ID 1073414.1).

Check Supported Database Server Requirements

General Requirements for a database server running SIM include:

Supported On	Versions Supported
Database Server OS	OS certified with Oracle Database 11gR2 Enterprise Edition. Options are: ▪ Oracle Linux 5 Update 3 (OL5.3) for x86-64 ▪ Red Hat Enterprise Linux 5 Update 3 (RHEL 5.3) for x86-64 ▪ IBM AIX 6.1 (Actual hardware or LPARs) ▪ Solaris 10 Sparc (Actual hardware or Logical Domains) ▪ HP-UX 11.31 Integrity (Actual hardware or HPVM)
Database Server	Oracle Database Enterprise Edition 11gR2 (11.2.0.1) with the following oneoff patches: ▪ 9582272: ORA-600[kkdlReadOnDiskDefVal: error] occurs when ALTER TRIGGER is executed. ▪ 9100882: SOA: ORA-600[KGHFRE3] SIGNALLED. ▪ 9010222: APPS ST 11G ORA-00600 [KKSFBC-REPARSE-INFINITE-LOOP] ▪ 9932143: [CTS] 3 FAILURES IN JMS/AQ RUN DIDN'T GET EXPECTED MSG BACK AND REDELIVERED FLAG ▪ 9130054: MASSIVE ORA-2051 SIGNALLED DURING SOA TEST AGAINST AN 11.2.0.1 DATABASE. ▪ 9367425: PROCESS CRASHED WHEN USING 11GR2 JDBC/OCI. ▪ 9495959: HANG WHEN TWO THREADS TRY TO CREATE THE ENV HANDLE AT THE SAME. **Note:** Patches 9367425 and 9495959 must be applied together. **Components:** ▪ Oracle Database 11g ▪ Oracle Partitioning ▪ Oracle Net Services ▪ Oracle Call Interface (OCI) ▪ Oracle Programmer ▪ Oracle XML Development Kit ▪ Examples CD (Formerly the companion CD) **Other components:** ▪ Perl compiler 5.0 or later ▪ X-Windows interface

Check Supported Application Server Requirements

The SIM application can be deployed on either Oracle WebLogic 10.3.3 or Oracle Application Server 10g 10.1.3.4.

> **Note:** If you are integrating with RMS 13.1.x products, then SIM 13.2.1 must be run on Oracle Application server (OAS).

General requirements for an Oracle Application Server capable of running the SIM application include the following.

> **Note:** Files required for OCM (Oracle Configuration Manager) are removed after OPatch is used to patch the WebLogic server. This will cause the product installers and OCM installation to fail. To work around this issue, back up the content of the $ORACLE_HOME/utils/ccr/lib directory prior to applying a patch using OPatch, and recopy the content back after you apply any patches. ORACLE_HOME is the location where WebLogic Server has been installed.
>
> **Note:** If using an OPatch on Linux 64-bit platforms, see Installer Fails because of missing .jar in $ORACLE_HOME/utils/ccr/lib in Appendix: Common Installation Errors.

Supported On	Versions Supported
Application Server OS	OS certified with Oracle Application Server 10g 10.1.3.4. Options are: ▪ Oracle Linux 5 Update 3 (OL5.3) for x86-64 ▪ Red Hat Enterprise Linux 5 Update 3 (RHEL 5.3) for x86-64 ▪ IBM AIX 6.1 (Actual hardware or LPARs) ▪ Solaris 10 Sparc (Actual hardware or Logical Domains) ▪ HP-UX 11.31 Integrity (Actual hardware or HPVM)
Application Server	Oracle Application Server 10g 10.1.3.4 with the following patches: ▪ 4601861 - NEED TO EXPOSE NZOS_SETIOSEMANTICS - Sun ▪ 5632264 - NEED UPDATED TIMEZONE FILES (VERSION 4) FOR MORE DST RULE CHANGES CORE - Generic Platform ▪ 5649850 - IF STRONG VERIFIER, GETCONNECTION FAIL AFTER INVOKE SETCONNECTIONCACHEPROPERTIES - Generic Platform (patch to help with uppercase passwords)

General requirements for an Oracle WebLogic Server capable of running the SIM application include the following.

Supported on:	Versions Supported:
Application Server OS	OS certified with Oracle Fusion Middleware 11g Release 1 (11.1.1.3). Options are: ▪ Oracle Linux 5 Update 3 (OL5.3) for x86-64 (Actual hardware or Oracle Virtual Machine) ▪ Red Hat Enterprise Linux 5 Update 3 (RHEL 5.3) for x86-64 (Actual hardware or Oracle Virtual Machine) ▪ IBM AIX 6.1 (Actual hardware or LPARs) ▪ Solaris 10 Sparc (Actual hardware or Logical Domains) ▪ HP-UX 11.31 Integrity (Actual hardware or HPVM)
Application Server	Oracle Fusion Middleware 11g Release 1 (11.1.1.3) **Components:** ▪ Oracle WebLogic Server 11g Release 1 (10.3.3) **Other components:** ▪ Oracle BI Publisher 10g (10.1.3.4) **Optional (SSO required)** ▪ Oracle Internet Directory 10g (10.1.4) ▪ Oracle WebTier 11g (11.1.1.3)

Note: This release of SIM is only supported in a managed OC4J instance as part of OracleAS 10g. It is not supported on OC4J standalone

Check Single Sign-On Requirements

If SIM is not being deployed in a Single Sign-On environment, skip this section.

If Single Sign-On is to be used, verify the Oracle Infrastructure Server 10g server has been installed. Verify the OAS HTTP server or Oracle HTTP Server for Web Logic server used to launch SIM has been registered with the Oracle Single Sign-On server.

For more details on this, see the *Oracle Single Sign-On Administration Guide*.

Check Directory Server Requirements

SIM uses directory server based user authentication and searching. For LDAP, SIM is certified with the following directory servers:

- Oracle Internet Directory 10.1.4

Check Third-Party Software Dependencies

- Oracle Retail Wireless Foundation Server, provided by Wavelink 4.x.

Check Client PC and Web Browser Requirements

Requirement	Versions
Operating system	Windows 2000 or XP
Display resolution	1024x768 or higher
Processor	1GHz or higher
Memory	512MBytes or higher
Java Runtime Environment	Oracle JRE 6u21-b08 or higher
Browser	Microsoft Internet Explorer 7, Microsoft Internet Explorer 8, or Mozilla Firefox 3.6.x The browser is used to launch the Java WebStart client.

Note: Oracle Retail does not recommend or support installations with less than 128 kb bandwidth available between the PC client and the data center. Limiting the client to less than 128 kb total available bandwidth causes unpredictable network utilization spikes, and performance of the client degrades below requirements established for the product. The 128 kb requirement provides reasonable, predictable performance and network utilization.

Supported Oracle Retail Products

The following Oracle Retail products can be integrated with SIM. Next to each product is an indication of whether it is required or optional for SIM to function properly:

- Retail Integration Bus (RIB) 13.2 and all subsequent patches and hot fixes – Required

  Although typically used to integrate SIM with RMS, RIB can also be used to integrate SIM with other merchandising systems.

  Note: RIB requires custom modifications to use a merchandising system other than RMS .

Retail Merchandising System (RMS) 13.2 – Optional

- Oracle Retail Price Management 13.2 – Optional
- Oracle Retail POS Suite 13.3 – Optional

The above products can be installed before or after SIM. However, it is helpful to know the connection details for the other products ahead of time so that you can provide them to the SIM application installer, which will configure the connection points for you.

Note: SIM 13.2.1 also support integration to RMS 13.1.x, RPM 13.1.x through RIB 13.1.x. The supported applications server for integrating with RIB13.1.x is OAS server.

UNIX User Account Privileges to Install the Software

A UNIX user account is needed to install the software. The UNIX user that is used to install the software should have write access to the WebLogic server installation files. For example, "oretail."

> **Note:** Installation steps will fail when trying to modify files under the WebLogic installation unless the user has write access.

SIM Installation Overview

The following basic steps are required to install and set up SIM.

1. Install the database (with or without RAC).
2. Install application server (WebLogic or OAS) if it has not been installed
3. Set role-based access control. See Chapter 2 of the *Oracle Retail Store Inventory Management Implementation Guide* for instructions.
4. Install the SIM application.
5. Run data-seeding from RMS.

Customer Profiles

SIM 13.2.1 supports multiple installation scenarios. Find your scenario below and proceed with the instructions listed for your scenario.

- If you are doing a fresh install of SIM 13.2.1 and have Oracle Retail Merchandising 13.1 applications OR ORPOS 13.2 then proceed with installing Oracle Application server 10g and Database install. That is, all these products run on Oracle Application server 10g.
- If you are doing a fresh install of SIM 13.2.1 and have Oracle Retail Merchandising 13.2 applications OR ORPOS 13.3 then proceed with installing Oracle Web logic server 11g and Database install. That is, all these products run on Oracle Web logic server 11g.
- If you are upgrading from SIM 13.2.0.3 to SIM 13.2.1, please see the *Oracle Retail Upgrade Guide* for database upgrade. Your choice of application server will be determined by the release levels of other Oracle Retail software with which SIM integrates.

2

RAC and Clustering

The Oracle Retail Store inventory Management System has been validated to run in two configurations on Linux:

- Standalone Oracle Application Server or Web Logic Server and Database installations
- Real Application Cluster Database and Oracle Application Server or Web Logic Server Clustering

The Oracle Retail products have been validated against an 11.2.0.1 RAC database. When using a RAC database, all JDBC connections should be configured to use OCI connections rather than THIN connections. It is suggested that when using OCI connections, the Oracle Retail products database be configured in the tnsnames.ora file used by the Oracle Application Server or Web Logic Server installations.

Clustering for Oracle Application Server 10.1.3 is managed as an Active-Active cluster accessed through a hardware Load Balancer. It is suggested that a VirtualHost be added to the OAS 10.1.3 reflecting the Virtual Server Name configured in the load balancer. It is also suggested that the OC4J select method be configured to prefer the use of local OC4J instances. The Oracle Retail products are currently not validated to be distributable at the application level in an OAS 10.1.3 cluster.

Clustering for Oracle Application Server 10.1.2 is managed as an Active-Active cluster accessed through a hardware Load Balancer. It is suggested that the Web Cache installation included with OAS 10.1.2 be configured to reflect all application server Mid-Tier installations. Validation has been completed utilizing a RAC 11.2.0.1 Oracle Internet Directory database with the OAS 10.1.2 cluster.

Clustering for Web Logic Server 10.3.3 is managed as an Active-Active cluster accessed through a Load Balancer. Validation has been completed utilizing a RAC 11.2.0.1 Oracle Internet Directory database with the Web Logic 10.3.3 cluster. It is suggested that a Web Tier 11.1.1.3 installation be configured to reflect all application server installations if SSO will be utilized.

References for Configuration:

- Oracle® Application Server High Availability Guide 10g Release 3 (10.1.3) Part Number B15977-02
- Oracle® Fusion Middleware High Availability Guide 11g Release 1 (11.1.1) Part Number E10106-09
- Oracle® Real Application Clusters Administration and Deployment Guide 11g Release 2 (11.2) Part Number E16795-08

3

Database Installation Tasks

Expand the SIM Database Schema Installer Distribution

To expand the SIM database schema installation distribution, complete the following steps.

1. Log in to the UNIX server as a user which has sufficient access to run sqlplus from the Oracle Database installation.
2. Create a new staging directory for the SIM database schema installer distribution (sim13dbschema.zip). There should be a minimum of 50 MB disk space available for the database schema installation files. This location is referred to as INSTALL_DIR for the remainder of this chapter.
3. Copy sim13dbschema.zip to <INSTALL_DIR> and extract its contents.

Create Tablespaces

Before you run the SIM database schema installer, make sure that the following tablespaces have been created in the database: RETEK_INDEX, RETEK_DATA, USERS, and LOB_DATA. Below are sample tablespace creation statements for these tablespaces. Oracle Retail recommends the use of locally managed tablespaces with automatic extent and segment space management. These tablespaces are not sized for a production environment!

```
CREATE TABLESPACE RETEK_INDEX DATAFILE
    '<datafile_path>/retek_index01.dbf'  SIZE 500M
    AUTOEXTEND ON NEXT 100M MAXSIZE 2000M
    EXTENT MANAGEMENT LOCAL
    SEGMENT SPACE MANAGEMENT AUTO
;
CREATE TABLESPACE RETEK_DATA DATAFILE
    '<datafile_path>/retek_data01.dbf'  SIZE 500M
    AUTOEXTEND ON NEXT 100M MAXSIZE 2000M
    EXTENT MANAGEMENT LOCAL
    SEGMENT SPACE MANAGEMENT AUTO
;
CREATE TABLESPACE USERS DATAFILE
    '<datafile_path>/users01.dbf'  SIZE 100M
    AUTOEXTEND ON NEXT 100M MAXSIZE 2000M
    EXTENT MANAGEMENT LOCAL
    SEGMENT SPACE MANAGEMENT AUTO
;
CREATE TABLESPACE LOB_DATA DATAFILE
    '<datafile_path>/lob_data01.dbf'  SIZE 50M
    AUTOEXTEND ON NEXT 100M MAXSIZE 2000M
    EXTENT MANAGEMENT LOCAL
    SEGMENT SPACE MANAGEMENT AUTO
;
```

Create the SIM Database User

To create the SIM database user, complete the following steps.

1. Connect to the database as a user that has privilege to create other users

   The user in the database which will own the SIM tables must be created prior to running the SIM database schema installer. A create_user.sql script has been provided that can be used for this:

   <INSTALL_DIR>/sim/dbschema/dbscripts/utility/create_user.sql

   The script takes three arguments on the command line in sqlplus: user name, password, and temporary tablespace.

   Example: SQL> @create_user.sql SIMUSER mypassword TEMP

2. Review this script and run it as a user with adequate permissions, such as SYSTEM.

Run the SIM Database Schema Installer

This installer installs the SIM database schema, compile SIM objects, insert SIM data, and produce the sim_dba.sql script.

1. Expand the sim13dbschema.zip distribution into <INSTALL_DIR> if nor already done.
2. Set the following environment variables:
   - Set the ORACLE_HOME to point to an installation that contains sqlplus. It is recommended that this be the ORACLE_HOME of the SIM database.
   - Set the PATH to: $ORACLE_HOME/bin:$PATH
   - Set the ORACLE_SID to the name of your database
   - Set the NLS_LANG for proper locale and character encoding

     Example: Export NLS_LANG=AMERICAN_AMERICA.UTF8
3. If you are using an X server such as Exceed, set the DISPLAY environment variable so that you can run the installer in GUI mode (recommended). If you are not using an X server, or the GUI is too slow over your network, unset DISPLAY for text mode.
4. Change the directory to the <INSTALL_DIR>. Run the install.sh script. This launches the installer. After installation is completed, a detailed installation log file is created: <INSTALL_DIR>/ sim/dbschema/logs/sim-install-db.<timestamp>.log.

   Note: Appendix A contains details on every screen and field in the database schema installer.
5. When the installer finishes it prints the values of the database SID and database schema user. Note these values as they are needed later when you run the SIM application installer.
6. The SIM database schema installer will produce a sim_dba.sql script which must be reviewed by a DBA and then run on the database server in order to complete the installation.

Resolving Errors Encountered During Database Schema Installation

If the database schema installer encounters any errors, it halts execution immediately and prints to the screen which SQL script it was running when the error occurred. It also writes the path to this script to the .dberrors file. When this happens, you must run that particular script using sqlplus. After you are able to complete execution of the script, delete the .dberrors file and run the installer again. You can run the installer in silent mode so that you do not have to retype the settings for your environment. See Appendix: Installer Silent Mode for information about silent mode.

See Appendix: Common Installation Errors for a list of common installation errors.

Subsequent executions of the installer will skip the SQL scripts which have already been executed in previous installer runs. This is possible because the installer maintains a **.dbhistory** file with a listing of the SQL scripts that have been run. If you have dropped the SIM schema and want to start with a clean installation, you can delete the .dbhistory file so that the installer runs through all of the scripts again. It is recommended that you allow the installer to skip the files that it has already run.

4

Application Installation

Application Server Deployment Options

SIM 13.2.1 supports two different application servers for deployment:

- Oracle WebLogic Server 11g Release 1 (10.3.3)
- Oracle Application Server 10g Enterprise Edition (10.1.3.4)

Your choice of application server is determined by the release levels of Oracle Retail software with which SIM integrates.

- You must use Oracle WebLogic Server 11g to integrate with
  - Oracle Retail Merchandising 13.2 applications using Oracle Retail Integration Bus (RIB) 13.2
  - Oracle Retail Point-of-Service 13.3
- You must use Oracle Application Server 10g to integrate with
  - Oracle Retail Merchandising 13.1 applications using Oracle Retail Integration Bus 13.1
  - Oracle Retail Point-of-Service 13.2

Your application installation steps will vary depending on which application server you are using. Perform your application installation using the appropriate procedure:

- See Chapter 5, Installing the SIM Application on Oracle Application Server (OAS).
- See Chapter 6, Installing the SIM Application on WebLogic.

5

Installing the SIM Application on Oracle Application Server (OAS)

Before proceeding you must install Oracle Application Server 10g 10.1.3.4 plus the patches listed in Chapter 1 of this document. The SIM application is deployed to an OC4J instance within the OracleAS 10g installation. It is assumed Oracle database has already been configured and loaded with the appropriate SIM schema for your installation.

Oracle Configuration Manager

The Oracle Retail OCM Installer packaged with this release installs the latest version of OCM and will be the first set of screens in both the OAS and WebLogic application installers.

The following document is available through My Oracle Support: *Oracle Configuration Manager Installer Guide* (ID 1071030.1).

This guide describes the procedures and interface of the Oracle Retail Oracle Configuration Manager Installer that a retailer runs near the completion of its installation process.

Access My Oracle Support at the following URL:

`https://support.oracle.com`

OCM Documentation Link

`http://www.oracle.com/technology/documentation/ocm.html`

Create a New OC4J Instance and Group for SIM

You can skip this section if you are redeploying to an existing OC4J group in Oracle Application Server 10.1.3.4.

The SIM application must be deployed to its own dedicated OC4J group. For instructions on how to create a new OC4J group and instance, see "Adding and Deleting OC4J Instances" in the "Reconfiguring Application Server Instances" chapter of the *Oracle Application Server Administrator's Guide*.

1. Log in to the server which is running your OracleAS 10g installation. Set your ORACLE_HOME environment variable to point to this installation.
2. Choose a name for the new OC4J instance and group.

   Example: sim-oc4j-instance

   Example: sim_group

   Create this OC4J instance and group as documented in the *Oracle Application Server Administrator's Guide*.

   Example:
   $ORACLE_HOME/bin/createinstance
   –instanceName sim-oc4j-instance –groupName sim_group

When prompted for the oc4jadmin password, provide the same administrative password you gave for the Oracle Application Server installation. All OC4J instances running Oracle Retail applications must have the same oc4jadmin password.

3. (**Linux only**) Increase memory for the new OC4J instance by modifying $ORACLE_HOME/opmn/conf/opmn.xml. Locate the OC4J instance you just created, and add the -XX:PermSize=256m -XX:MaxPermSize=512m -Xms256m -Xmx256m options to the start-parameters section.

   Example:

   ```
   <process-type id="orco-inst" module-id="OC4J"
   status="enabled">
      <module-data>
         <category id="start-parameters">
            <data id="java-options" value="-server
   -XX:PermSize=256m -XX:MaxPermSize=512m -Xms256m -
   Xmx256m -
   Djava.security.policy=$ORACLE_HOME/j2ee/orco-
   inst/config/java2.policy –Djava.awt.headless=true
   –Dhttp.webdir.enabled=false"/>
         </category>
   ```

4. Force OPMN to reload the configuration file.

   Example: $ORACLE_HOME/opmn/bin/opmnctl reload

5. Start the OC4J group. You can do this through the Enterprise Manager Web interface, or on the command line using the opmnctl utility:

   Clustered Example: $ORACLE_HOME/opmn/bin/opmnctl @cluster startproc ias-component=sim_group

   Non-clustered Example:
   $ORACLE_HOME/opmn/bin/opmnctl startproc ias-component=sim_group

6. Verify that the OC4J group was fully started. If you are using the Enterprise Manager Web interface, the instance should have a green arrow indicating that it is running. On the command line, verify that the instance has a status of Alive.

   Example: $ORACLE_HOME/opmn/bin/opmnctl status

   If you are unable to start the OC4J instance after several attempts, try increasing the startup timeouts in ORACLE_HOME/opmn/conf/opmn.xml. If that does not help, consult the Oracle Application Server documentation for further assistance.

Configure Apache for JNLP Files

If this is the first WebStart application that is being installed in the HTTP server, you need to configure the **mime.types** file with the jnlp file type. If you are using the Apache distribution that is included with OracleAS, this file can be found under ORACLE_HOME/Apache/Apache/conf. Add the following line to the file:

```
application/x-java-jnlp-file          jnlp
```

Restart the Apache server for this change to take effect. If you do not add this line then jnlp files are served as plain text and you cannot launch the application.

Example: $ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=HTTP_Server

Set the LANG Environment Variable

The LANG environment variable must be set in the profile of the UNIX user who owns the application server ORACLE_HOME files. If you change the value of LANG or set the value for the first time, you must restart the Application Server in order for the change to take effect.

Example:

LANG=en_US

export LANG

For instructions on how to restart the Application Server, see the *opmnctl Commands* chapter of the *Oracle® Process Manager and Notification Server Administrator's Guide*.

Example:

$ORACLE_HOME/opmn/bin/opmnctl stopall

$ORACLE_HOME/opmn/bin/opmnctl startall

Expand the SIM Application Distribution

To expand the SIM application distribution, complete the following steps.

1. Log into the UNIX server as the user who owns the OracleAS 10g installation. Create a new staging directory for the SIM application distribution (sim13application.zip). There should be a minimum of 800 MB disk space available for the application installation files.

   Example: $ORACLE_HOME/j2ee/sim-oc4j-instance/sim-staging

   This location is referred to as INSTALL_DIR for the remainder of this chapter.
2. Copy sim13application.zip to <INSTALL_DIR> and extract its contents.

   Example: unzip sim13application.zip

Run the SIM Application Installer

This installer configures and deploys the SIM application and Java WebStart client files.

1. If you are installing to a clustered Application Server, perform the preinstallation tasks as described in the Clustered Installations -- Preinstallation Steps section above.
2. Set the ORACLE_HOME and JAVA_HOME environment variables. ORACLE_HOME should point to your OracleAS installation. JAVA_HOME should point to $ORACLE_HOME/jdk.
3. If you are using an X server such as Exceed, set the DISPLAY environment variable so that you can run the installer in GUI mode (recommended). If you are not using an X server, or the GUI is too slow over your network, unset DISPLAY for text mode.
4. Verify that the OC4J instance(s) that you install SIM to are currently running. See the section "Create a New OC4J Instance and Group for SIM" for instructions on how to start the oc4j instance(s).

5. Run the install.sh script. This launches the installer.
6. After installation is completed, a detailed installation log file is created: <INSTALL_DIR>/sim/application/logs/sim-install-app.<timestamp>.log.

> **Note:** Appendix B contains details on every screen and field in the Oracle Application Server application installer.

> **Note:** Appendix E contains details on common installation errors.

Review and/or Configure Oracle Single Sign-On

Skip this section if you are not using Single Sign-On for user identification and authentication.

Single Sign-On is applicable only to the JnlpLaunch Servlet. The JnlpLaunch Servlet is a dynamically protected application. The JnlpLaunch Servlet causes the SIM client application to execute under the SSO user name with a temporary password.

> **Note:** The JnlpLaunch servlet may be configured for either an SSO or non-SSO environment.

HTTP Server configuration requirements: The HTTP Server must be registered with the Oracle Single Sign-On server and the mod_osso module enabled. The registration process typically involves running the ssoreg.sh script at the OSSO server installation and copying the output osso.conf file to the HTTP Server. This process is documented in the Oracle Single Sign-On administration documentation.

JnlpLaunch requirements: The JnlpLaunch Servlet uses the configuration file, JnlpLaunch.properties, to control its behavior. Due to security considerations, this file must not be published or readable to the general public.

JnlpLaunch.properties has the following configuration entries that apply to Single Sign-On:

- `secret.key – Used` to create the temporary password, this property should contain a random string. If JnlpLaunch is deployed in a different JVM than the SIM Server EJBs, this string must be an exact match between the JnlpLaunch Servlet and the one available to the SIM EJBs. For security purposes, each separate instance of the SIM application (for example, test versus development) should have a different secret key.
- `user.validation.timeout` – Number of seconds the SIM Server uses to determine if a temporary password is still valid.
- `osso.used` – Determines if the JnlpLaunch Servlet will throw a 499 error when an unathenticated user has been detected. This property must be set to true if Oracle Single Sign-On is used and false if not.

The JnlpLaunch.properties file is initialized by the SIM installer and should contain valid entries for SSO when the Enable Single Sign-On in SIM? prompt was answered with Y or Yes. However, an administrator may want to alter the user.validation.timeout or other property after the initial installation.

SIM Batch Scripts

The SIM application installer places the SIM batch programs with the rest of the SIM application files under $ORACLE_HOME/j2ee/<oc4j-instance-name>/<sim-client-deployment-name>/batch

The batch programs can be run from a different location if you cannot run them from under the application server ORACLE_HOME. To install the batch files in a different location just copy the entire $ORACLE_HOME/j2ee/<oc4j-instance-name>/<sim-client-deployment-name>/batch directory to the appropriate destination.

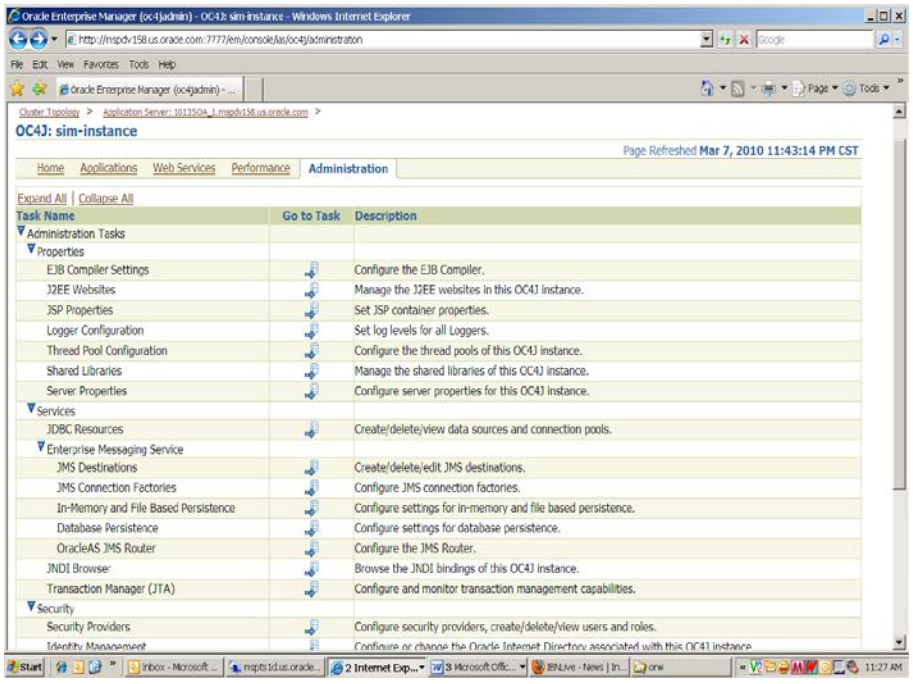
The batch location is assumed to be located on the same server as the application server. If you copy the batch to a location on a different server, then you need to configure the file path to the sim-batch.log file, which is defined in batch /resources/log4j.xml.

See the Batch Detail section of the *Oracle Retail Store Inventory Management Operations Guide* for information on how to run batches.

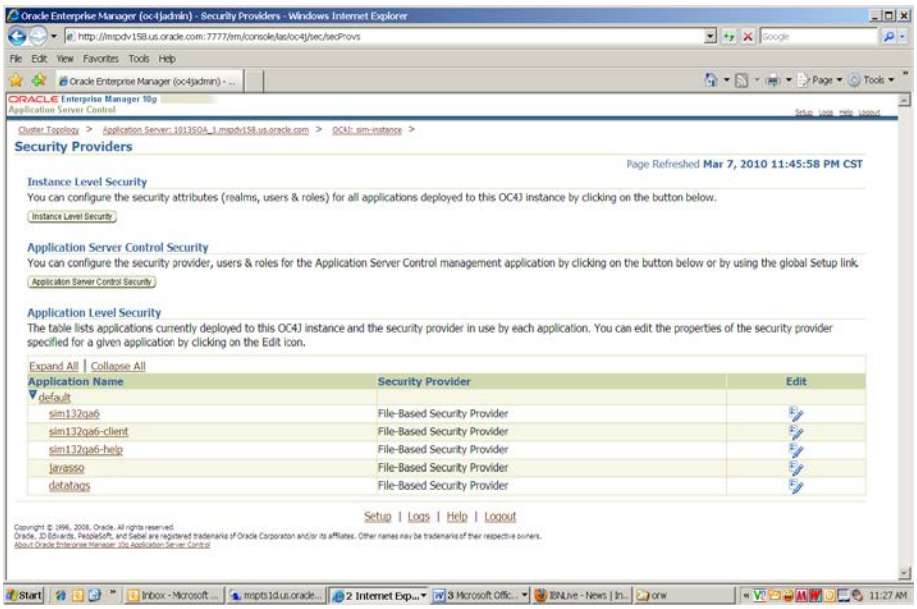
Adding Users to Application Server for Web Services

Once the application has been installed, you need to add users to the user role for web services.

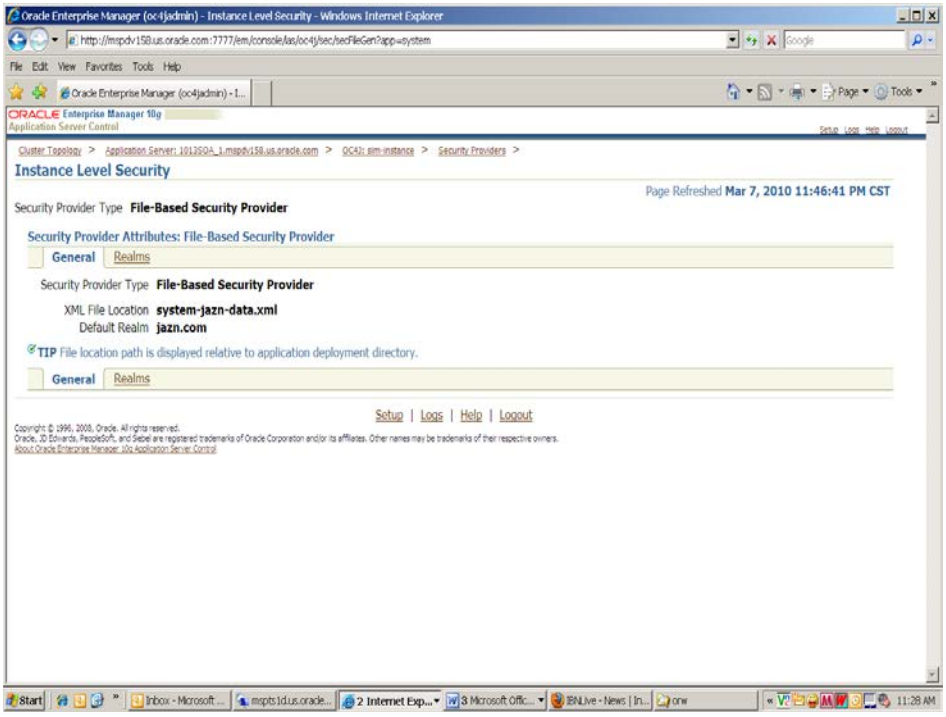
1. Go to the Enterprise Manager console for the Application Server where you installed SIM.
2. Click the SIM instance where you installed the application.
3. Click the Administration link.
4. Click the **Security Provider** task icon in the Security category.



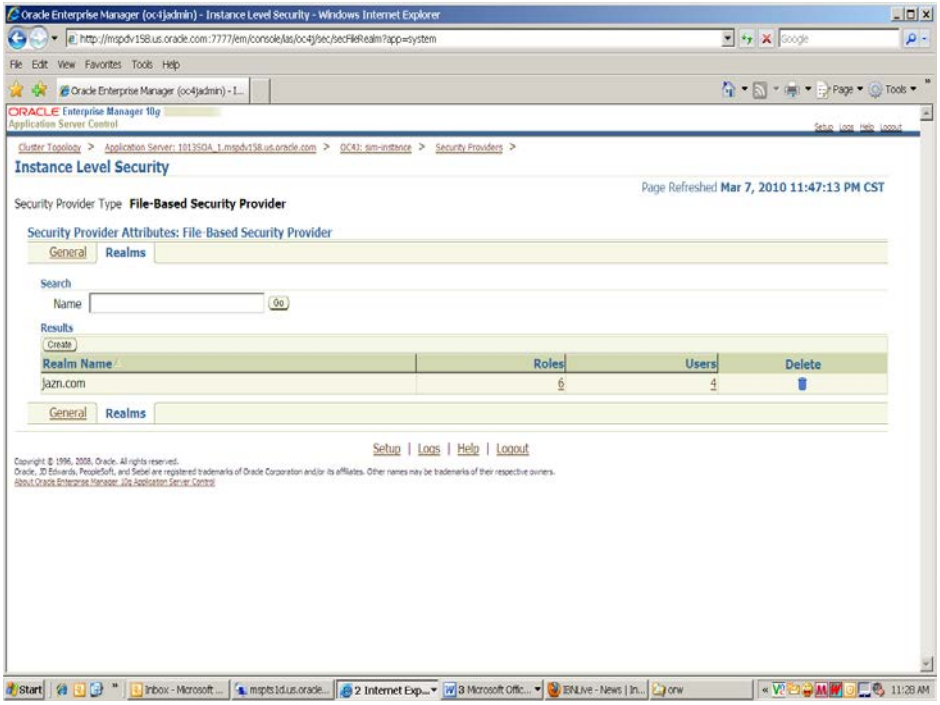
5. Click **Instance Level Security**.



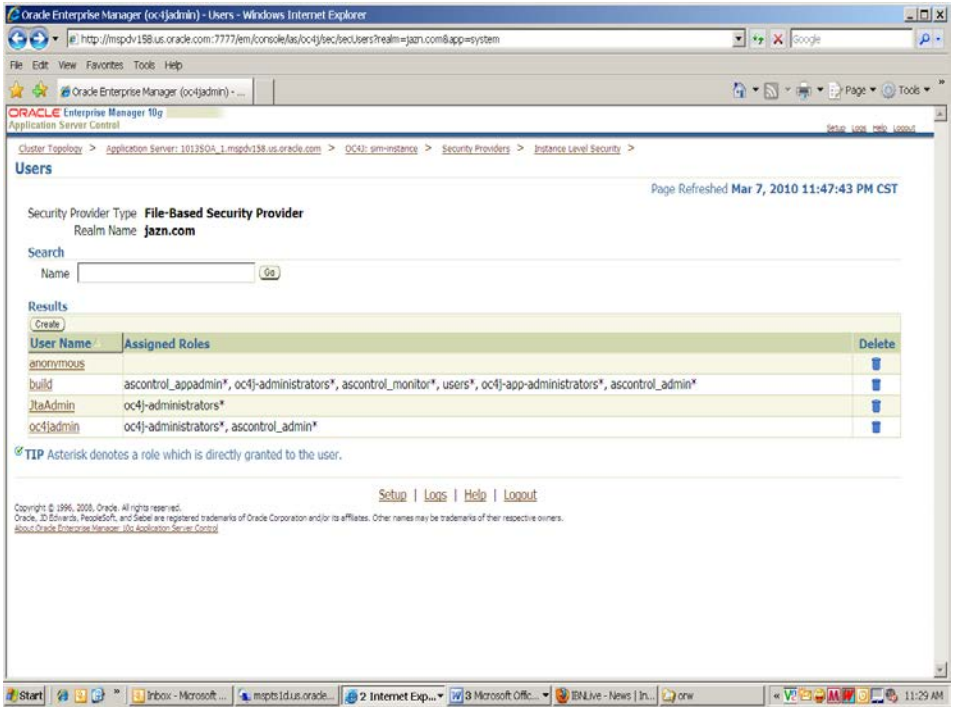
6. Click **Realms**.



7. Click **Users** (click the number under the Users column).



8. Create the user by clicking the **Create** button:



9. Create the user by adding user name, password, choosing the user role. Click **OK**.

Resolving Errors Encountered During Application Installation

If the application installer encounters any errors, it halts execution immediately. You can run the installer in silent mode so that you do not have to retype the settings for your environment. See Appendix: Installer Silent Mode for information about silent mode.

See Appendix: Common Installation Errors for a list of common installation errors.

Since the application installation is a full reinstall every time, any previous partial installs are overwritten by the successful installation.

Backups Created by Installer

The SIM application installer backs up the sim-home directory if it finds a previous installation of SIM. The backups are made by adding a suffix to the file or directory with a timestamp. This is done to prevent the removal of any custom changes you might have. These backup directories can be safely removed without affecting the current installation.

Example: sim-home-backup-200708171550

Web Help Files

The application installer automatically copies the web help files to the proper location. They are accessible from the help links within the application.

Starting and Stopping SIM

SIM can also be restarted by using the Enterprise Manager to restart the OC4J instance that contains SIM. However, if you use the Enterprise Manager to restart SIM, the Wavelink server needs to be restarted manually.

Starting and Stopping the Wavelink Server

In order to use handheld wireless devices with SIM, the Wavelink server must be running.

Note: If you use the Enterprise Manager to restart SIM, then you need to restart the Wavelink server manually.

If you use SIM's startup and shutdown scripts to restart SIM on the command line, then the Wavelink server will also be restarted along with SIM. However, if you use the Enterprise Manager to restart SIM, the Wavelink server is not affected. So it must be restarted manually once SIM is running again.

The Wavelink server scripts can be found here:

ORACLE_HOME/ j2ee/<oc4j-instance-name>/<sim-deployment-name>/wireless/bin/wavelink-startup.sh

ORACLE_HOME/ j2ee/<oc4j-instance-name>/s<sim-deployment-name>/wireless/bin/wavelink-shutdown.sh

Note: The wireless functionality in SIM is dependent on Wavelink and includes a client and server component. Wavelink software ensures that the wireless user interface of SIM can work with various handheld devices.

For the handheld to interact correctly with SIM, it is required to install the appropriate Wavelink studio client. The Wavelink studio client and its installation instructions can be found at http://www.wavelink.com/download/downloads.aspx.

The Oracle Retail Wireless Foundation Server is bundled with the SIM server. It has a single session free license. For multiple sessions additional licenses need to be obtained.

Please contact your Oracle sales representative or client partner for Wavelink Studio Client and Oracle Retail Wireless Foundation Server license information.

Note: For configurations of physical handheld devices or wireless network setup, check your hardware manufacturer's manual or Wavelink's studio client information. This information is not covered in this guide.

Note: For additional information about LDAP configuration see the *Oracle Retail Store Inventory Management Implementation Guide*.

6

Installing the SIM Application on WebLogic

These instructions apply to new installations. Before proceeding, you must install Oracle WebLogic Server 11g Release 1 (10.3.3) and patches listed in the Chapter 1 of this document. The Oracle Retail Allocation application is deployed to a WebLogic Managed server within the WebLogic installation. It is assumed Oracle RDBMS has already been configured and loaded with the appropriate RMS and Oracle Retail Allocation schemas for your installation.

If Oracle Forms 11g has been installed in the same WebLogic that is being used for this application, a domain called "ClassicDomain" is installed. Installing a separate domain under the same WebLogic server is recommended. It can be called "APPDomain" (or something similar) and will be used to install the non-ORACLE Forms managed servers. Applications such as RPM, SIM, Allocation, ReIM, RIB, AIP, and RSL can be installed in the "APPDomain."

Oracle Configuration Manager

The Oracle Retail OCM Installer packaged with this release installs the latest version of OCM and will be the first set of screens in both the OAS and WebLogic application installers.

The following document is available through My Oracle Support (formerly MetaLink). Access My Oracle Support at the following URL:

`https://metalink.oracle.com`

Oracle Configuration Manager Installer Guide (Doc ID: 1071030.1)

This guide describes the procedures and interface of the Oracle Retail Oracle Configuration Manager Installer that a retailer runs near the completion of its installation process.

OCM Documentation Link

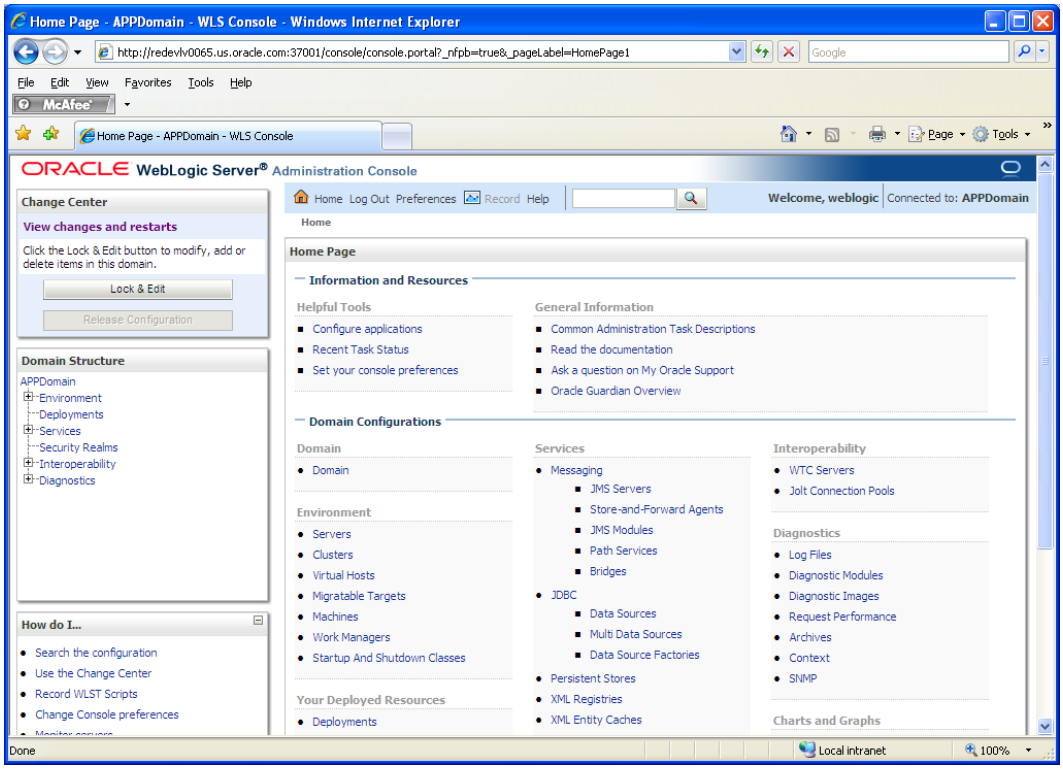
`http://www.oracle.com/technology/documentation/ocm.html`

Install Managed Server in WebLogic

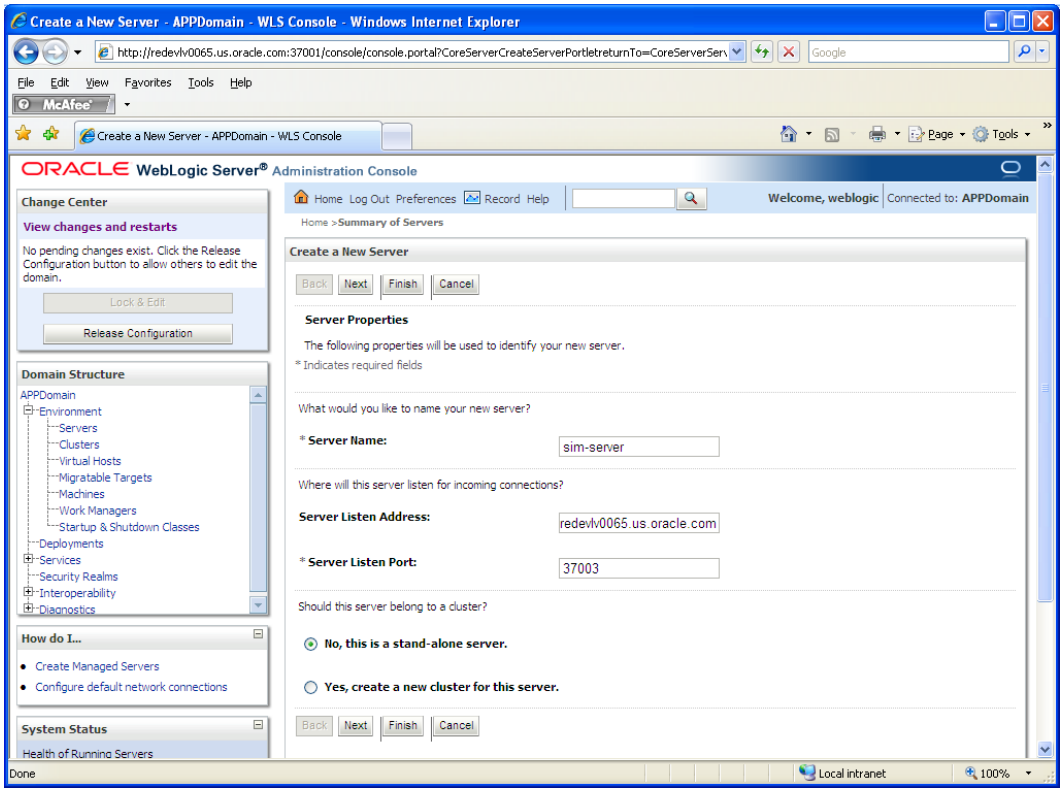
Before running the application installer, you must install the managed server in WebLogic if it was not created during the domain install.

> **Note:** If integrating SIM with RSL, having SIM and RSL servers configured in the same domain is recommended. If RSL server is installed in a different domain, you must setup a "trusted relationship" between the two WebLogic domains due to RMI calls.

1. Log in to the admin console



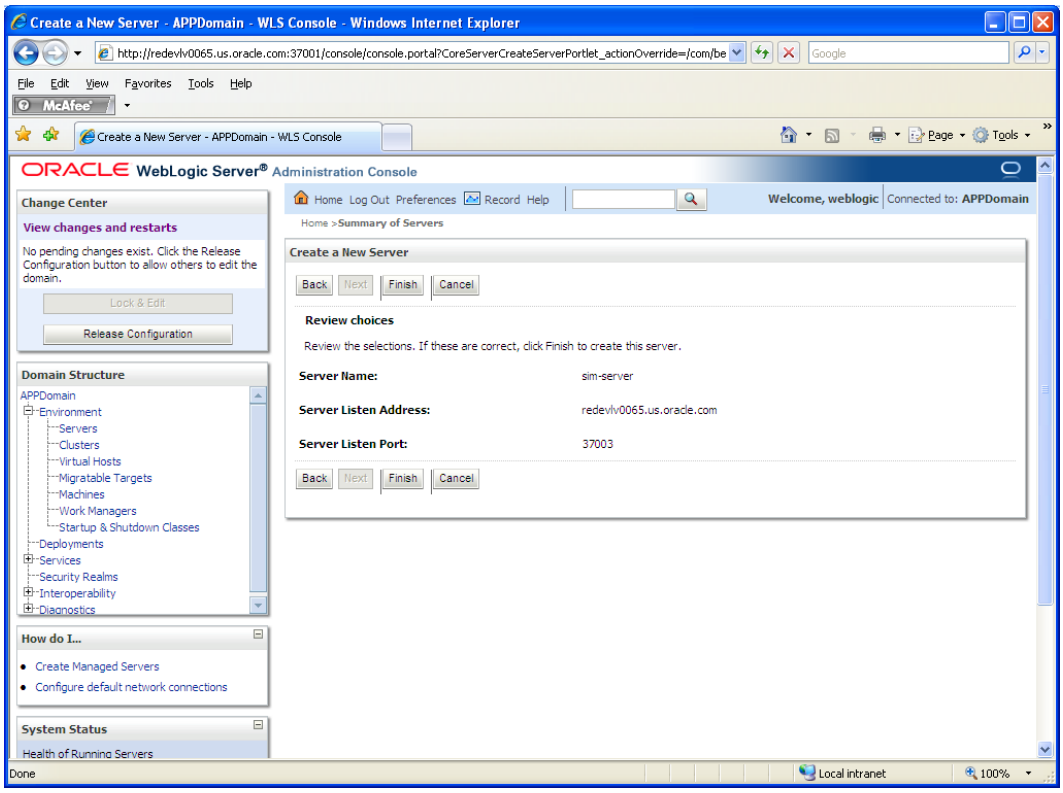
2. Click **Lock & Edit**.
3. Navigate to Environment->Servers and select new tab of the servers on the right side.



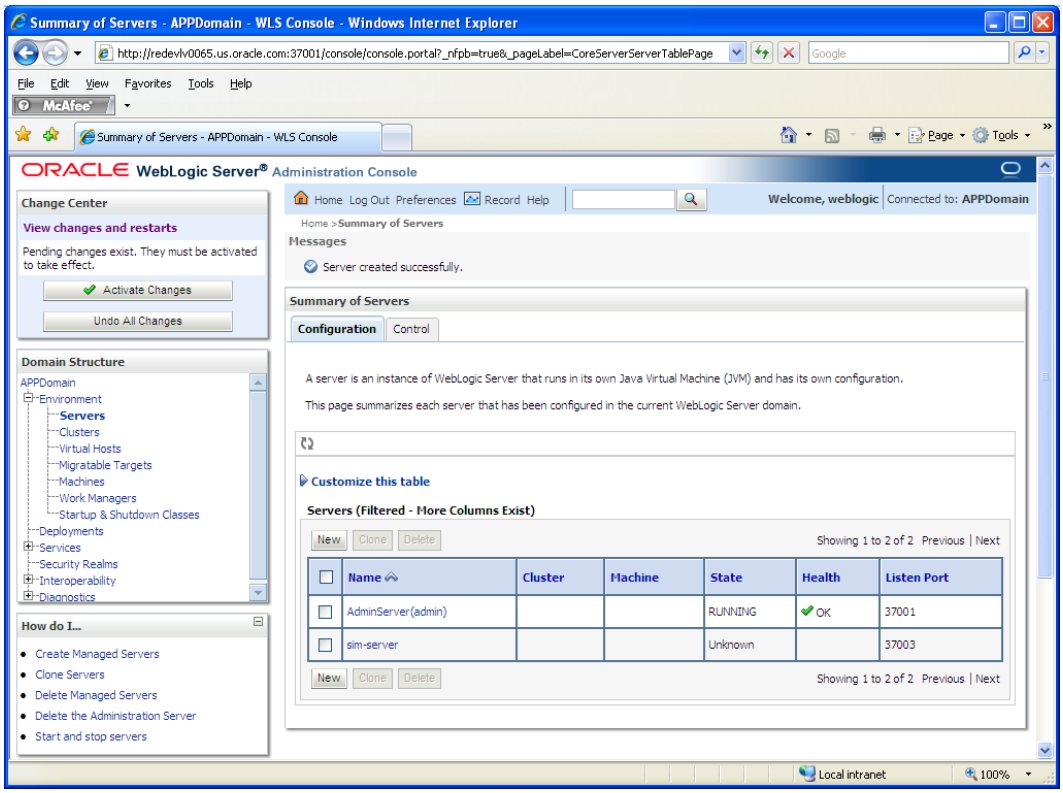
4. Set the following variables:
   - **Server Name**: These should be some name specific to your application targeted

     Example: sim-server
   - **Server Listen Address**: <weblogic server> (for example, redevlv0065.us.oracle.com).
   - **Server Listen Port**: A free port; you should check for availability.

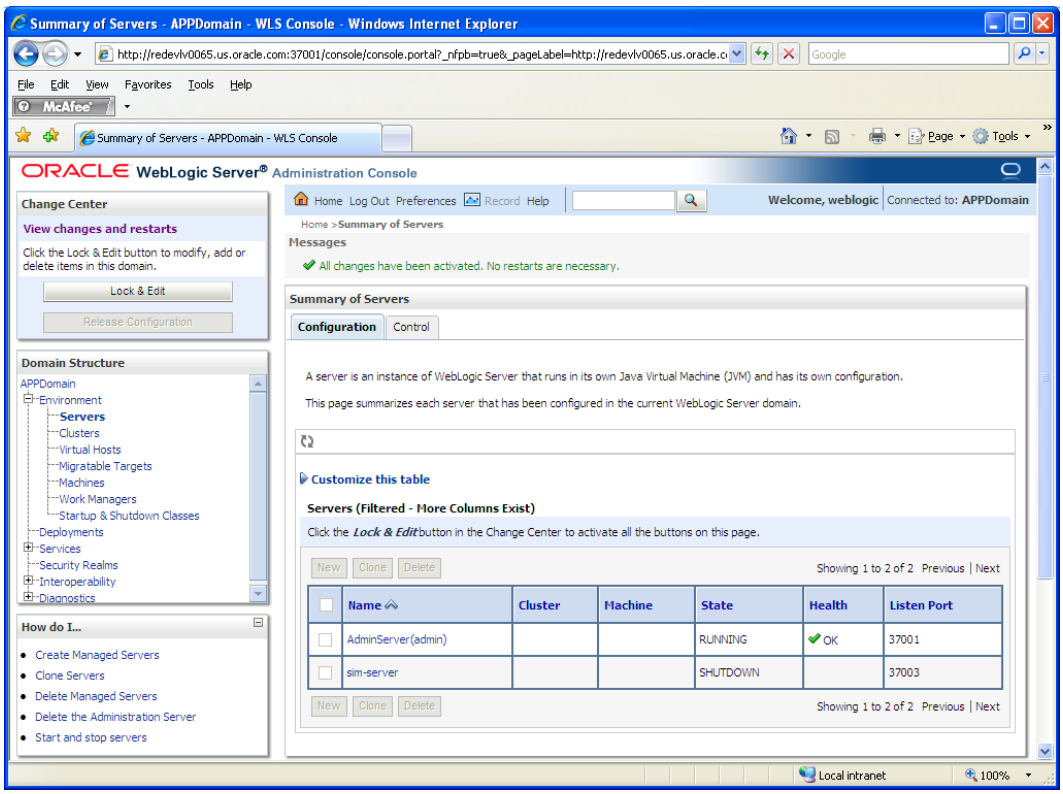
     A suggestion is to increment the AdminServer port by two and keep incrementing by two for each managed server (for example, 18003, 18005, 18007 and so on).
5. Click **Next**.



6. Click **Finish**.



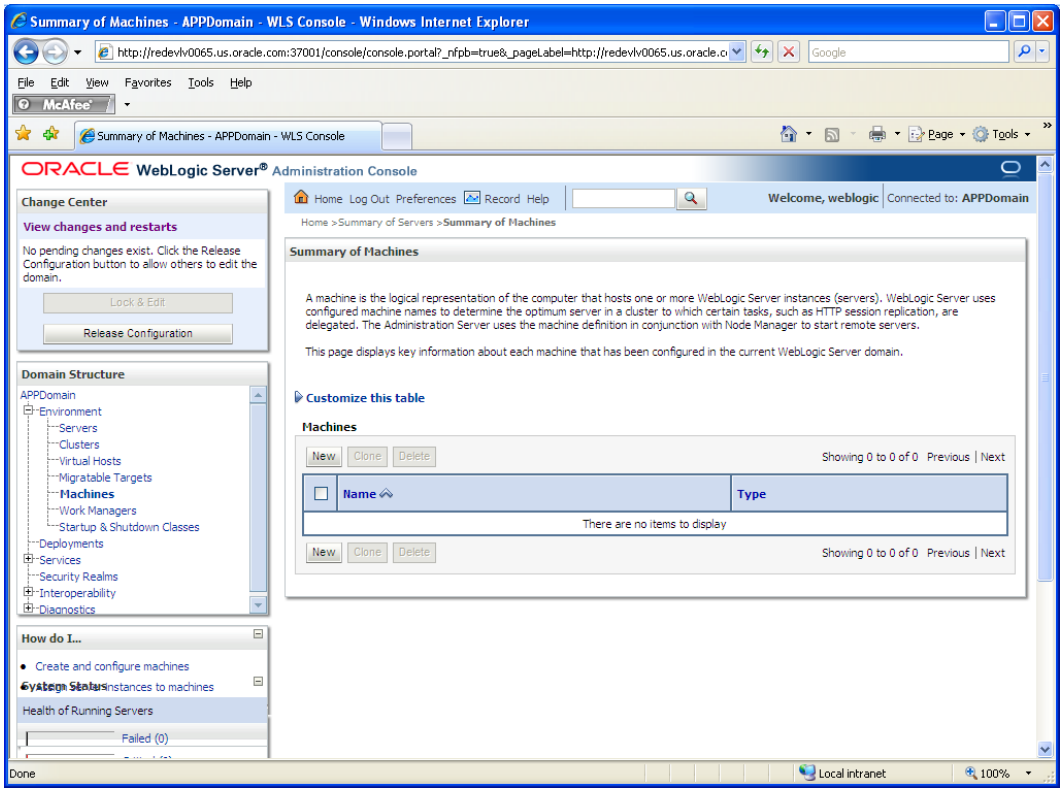
7. Click **Activate Changes** on the left hand side.



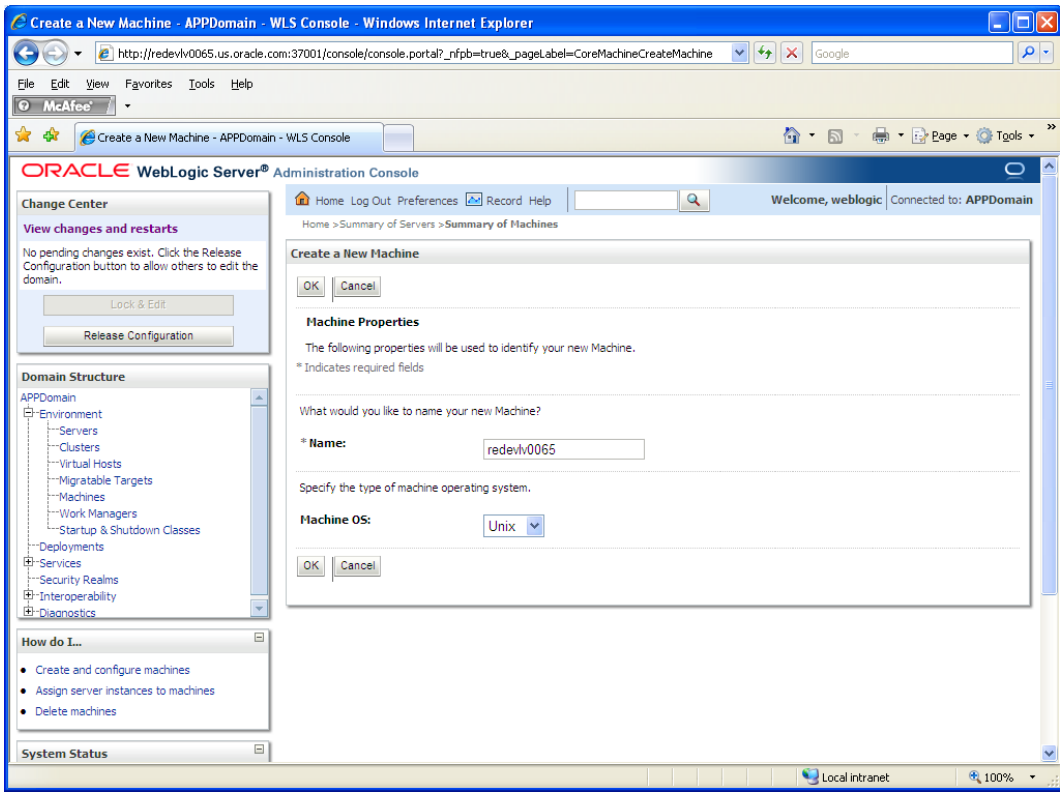
Install NodeManager

Install NodeManager if it was not created during domain install. NodeManager is required so that the managed servers can be started and stopped via the admin console. Only one NodeManager is needed per WebLogic install.

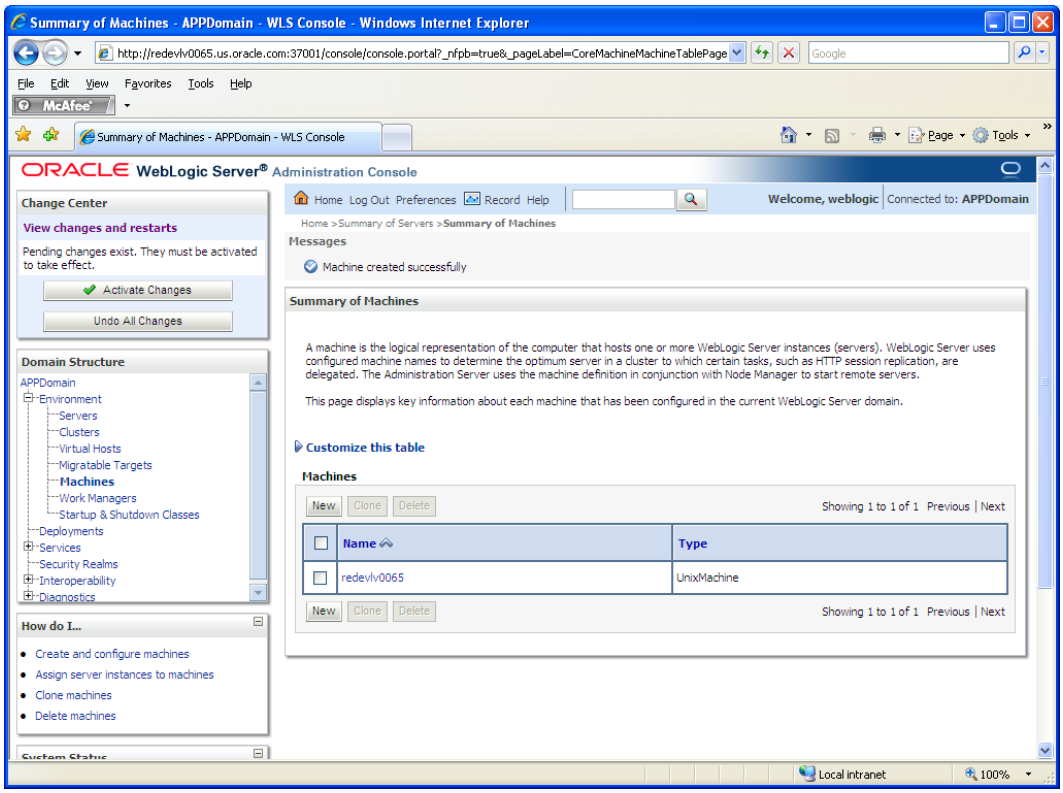
1. Log in to the admin console.
2. Click **Lock & Edit** and navigate to Environments > Machines.



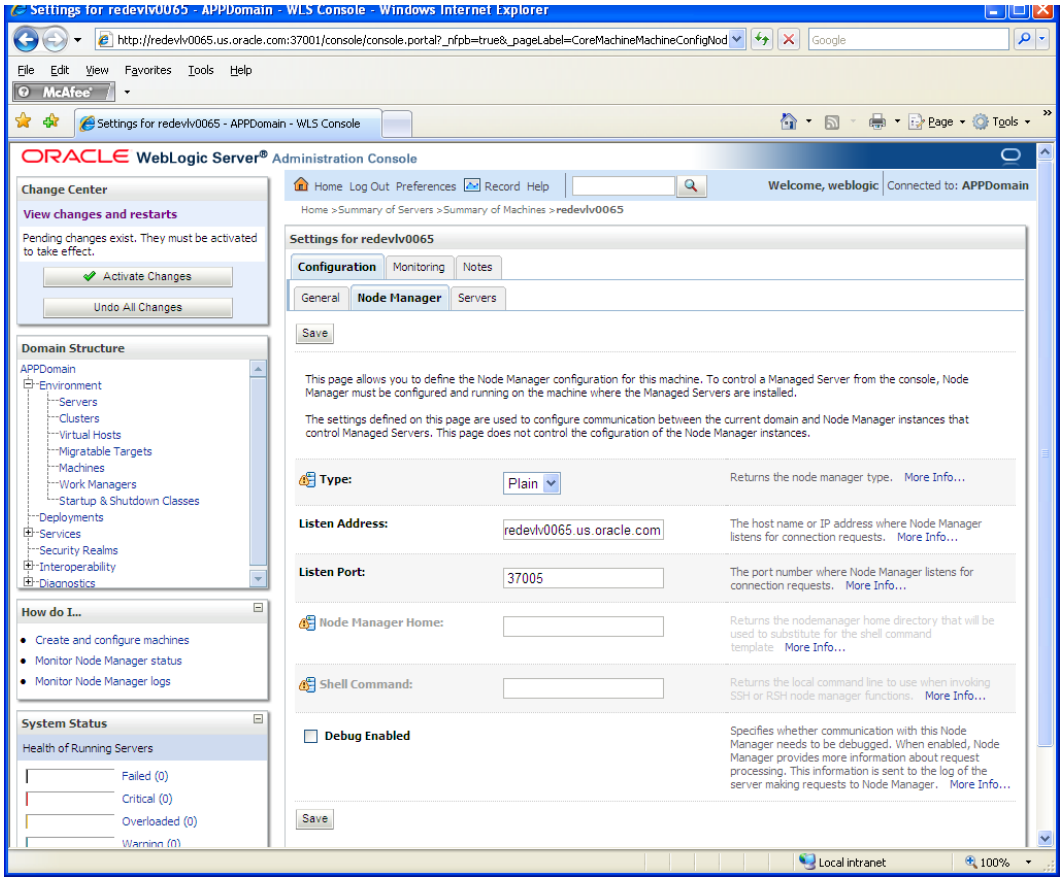
3. Click **New**.



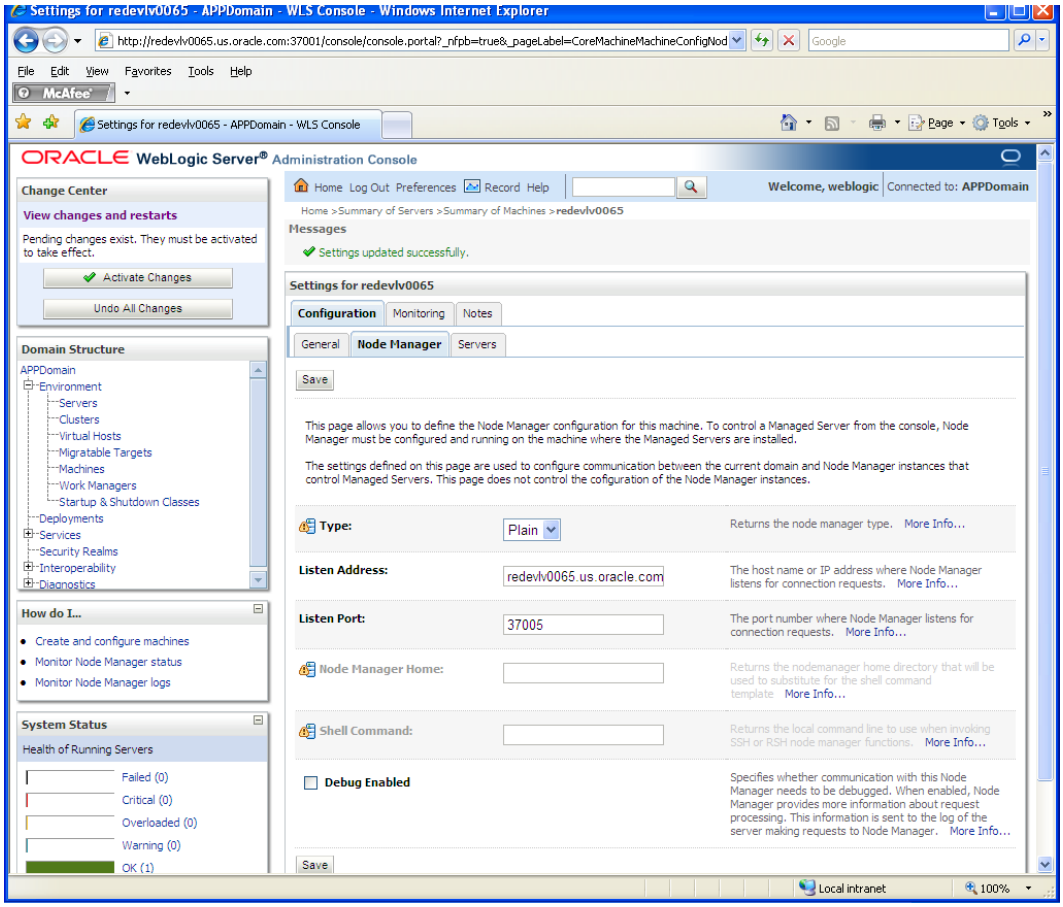
4. Set the following variables:
   - **Name**: Logical machine name
   - **Machine OS**: UNIX
5. Click **OK**.
6. Click on the machine created below.



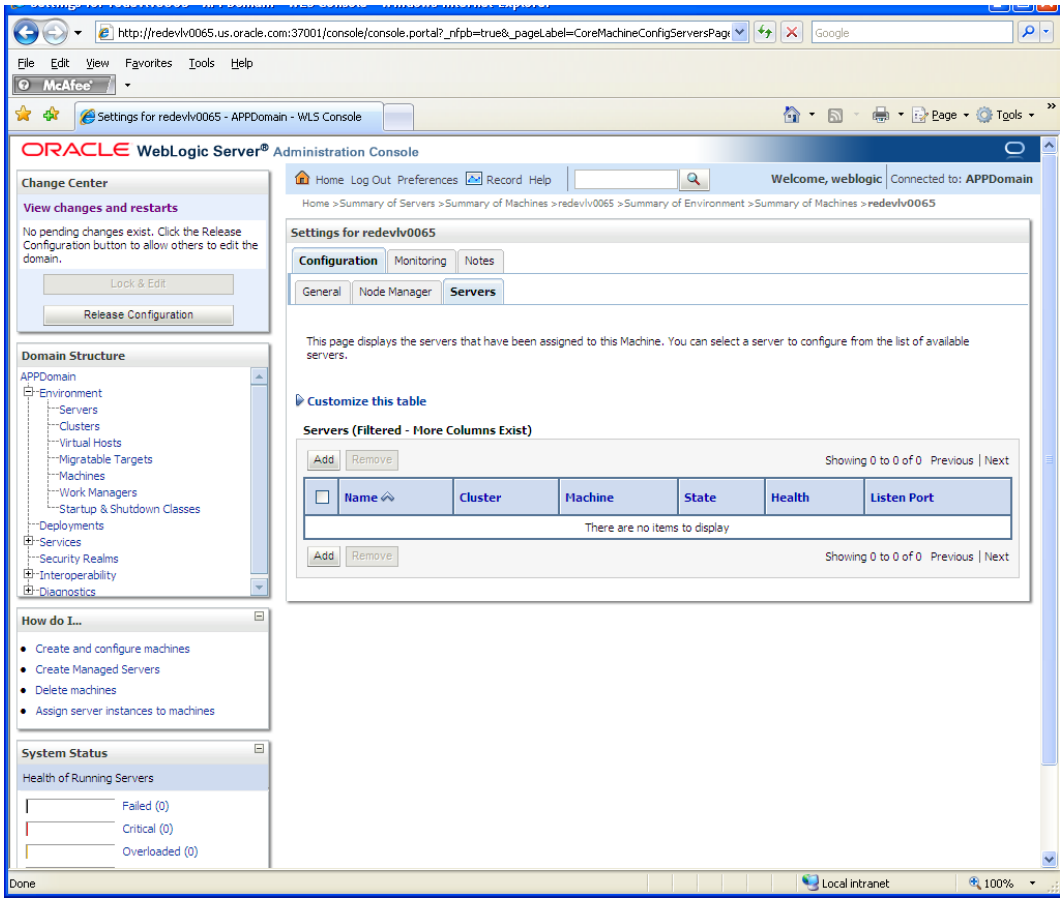
7. Click on the NodeManager tab and update the details below.
   - **Type**: Plain
   - **Listen Address**: redevlv0065.us.oracle.com
   - **Listen Port**: NodeManager will be assigned a default port (for example, 5556)


Settings for redevlv0065 - APPDomain - WLS Console - Windows Internet Explorer
http://redevlv0065.us.oracle.com:37001/console/console.portal?_nfpb=true&_pageLabel=CoreMachineMachineConfigNod
File Edit View Favorites Tools Help
Settings for redevlv0065 - APPDomain - WLS Console
ORACLE WebLogic Server® Administration Console
Change Center
View changes and restarts
Pending changes exist. They must be activated to take effect.
Activate Changes
Undo All Changes
Domain Structure
APPDomain
Environment
Servers
Clusters
Virtual Hosts
Migratable Targets
Machines
Work Managers
Startup & Shutdown Classes
Deployments
Services
Security Realms
Interoperability
Diagnostics
How do I...
Create and configure machines
Monitor Node Manager status
Monitor Node Manager logs
System Status
Health of Running Servers
Failed (0)
Critical (0)
Overloaded (0)
Home Log Out Preferences Record Help
Welcome, weblogic Connected to: APPDomain
Home >Summary of Servers >Summary of Machines >redevlv0065
Settings for redevlv0065
Configuration Monitoring Notes
General Node Manager Servers
Save
This page allows you to define the Node Manager configuration for this machine. To control a Managed Server from the console, Node Manager must be configured and running on the machine where the Managed Servers are installed.
The settings defined on this page are used to configure communication between the current domain and Node Manager instances that control Managed Servers. This page does not control the configuration of the Node Manager instances.
Type: Plain
Returns the node manager type. More Info...
Listen Address: redevlv0065.us.oracle.com
The host name or IP address where Node Manager listens for connection requests. More Info...
Listen Port: 37005
The port number where Node Manager listens for connection requests. More Info...
Node Manager Home:
Returns the nodemanager home directory that will be used to substitute for the shell command template More Info...
Shell Command:
Returns the local command line to use when invoking SSH or RSH node manager functions. More Info...
Debug Enabled
Specifies whether communication with this Node Manager needs to be debugged. When enabled, Node Manager provides more information about request processing. This information is sent to the log of the server making requests to Node Manager. More Info...
Save
Local intranet
100%

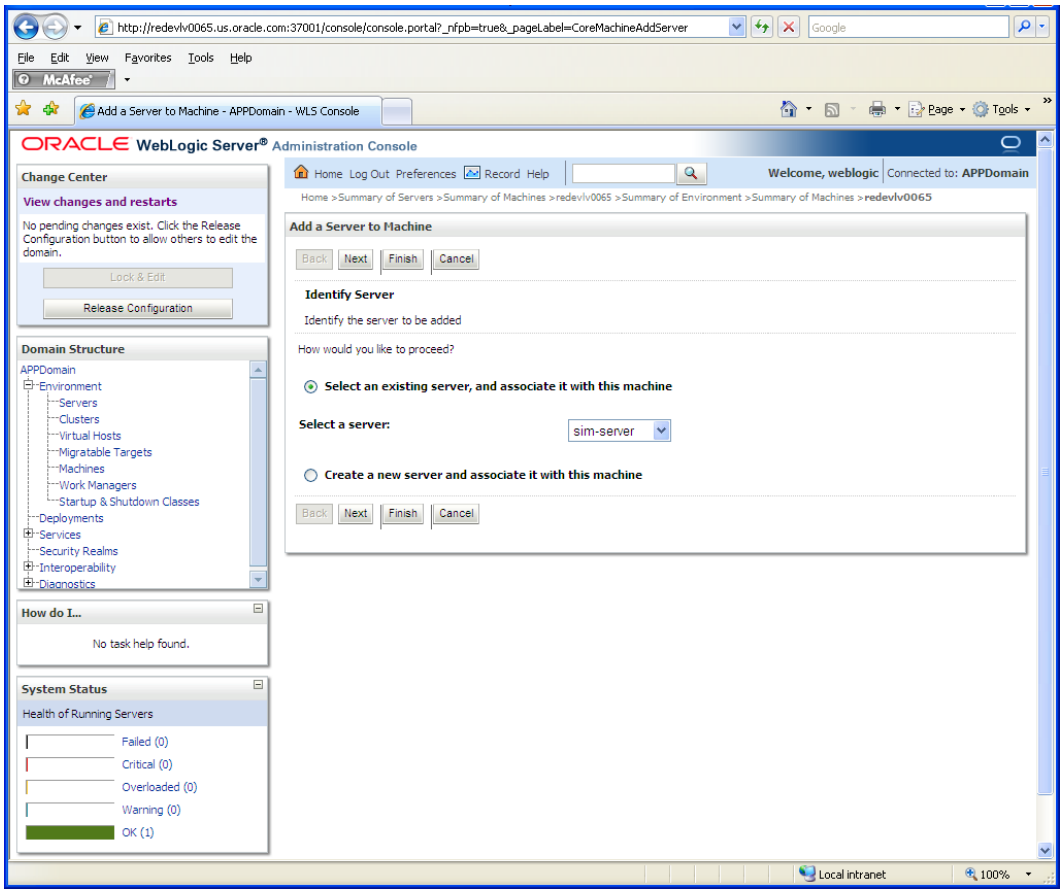
8. Click **Save**.



9. Click **Activate Changes**.
10. Click **Lock & Edit**.
11. Navigate to Environments > machines. Click on the machine name. Select the Servers tab.



12. Click **Add**. Add the managed servers that need to be configured with NodeManager.



13. Set the following variables:
    - Server: <sim-server>
14. Click **Next.** Click **Finish**.
15. Click **Activate Changes**.

> **Note:** To activate changes the server must be stopped:
>
> $WLS_HOME/user_projects/domains/<AppDomain>/bin /stopManagedWebLogic.sh adf_MS ${server_name}:${server_port}
>
> Go to each managed server that is being added to the machine and click the Server Start tab. In the Class Path box, add the following:
>
> `<full-path-to-domain>/servers/<managed-server>`
>
> For example:
> `/u00/webadmin/product/10.3.3_RIB/WLS_64/user_projects/domains/APPDomain/servers/sim-server`

16. After the CLASSPATH changes are finished, click **Save.**
17. Click **Activate Changes**.
18. Start NodeManager from the server using the startNodeManager.sh at $WLS_HOME/wlserver_10.3/server/bin.

19. Update nodemanager.properties file at the following location and set the SecureListener variable to false.
    $WLS_HOME/wlserver_10.3/common/nodemanager/nodemanager.properties

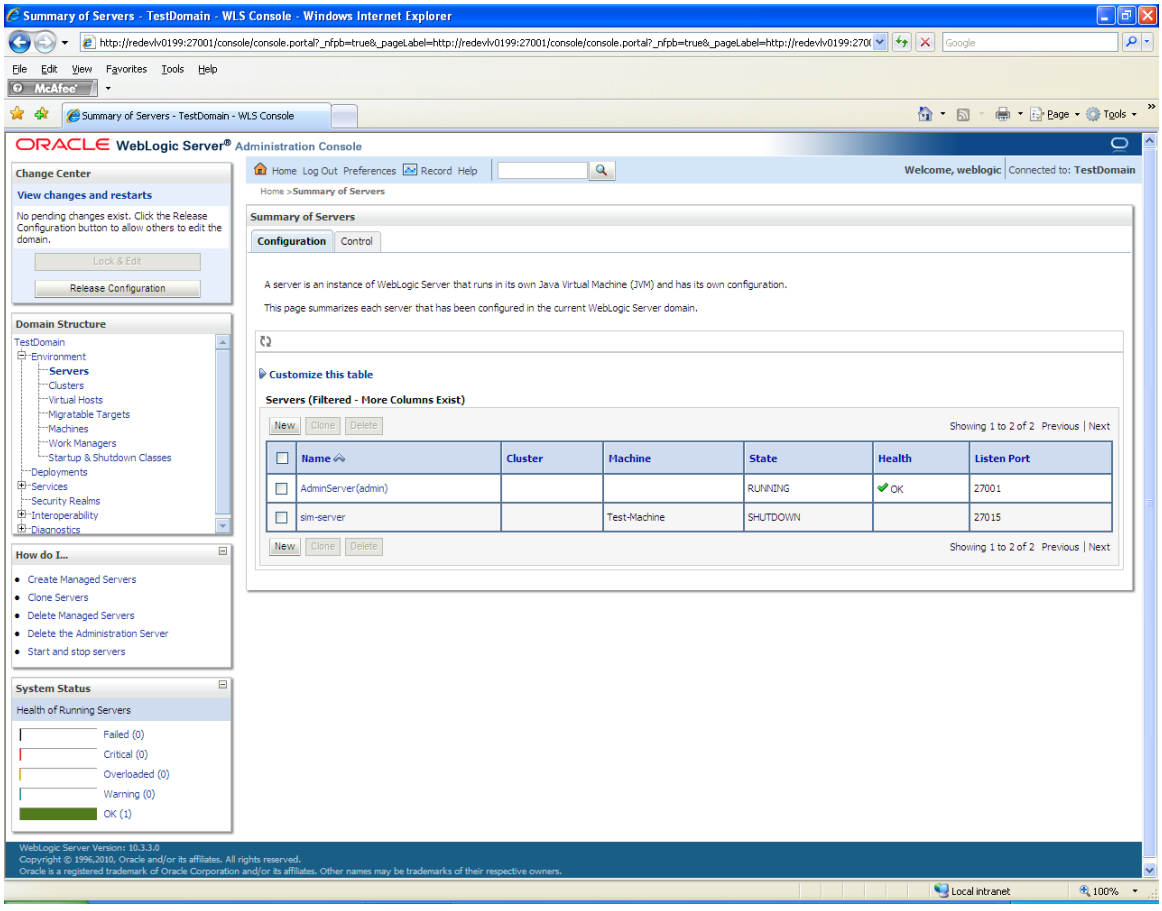
    SecureListener=false
20. After making changes to the nodemanager.properties file, NodeManager must be restarted.

> **Note:** The nodemanager.properties file is created after NodeManager is started for the first time. It is not available before that point.

Start the Managed Servers

After NodeManager is started, the managed servers can be started via the admin console.

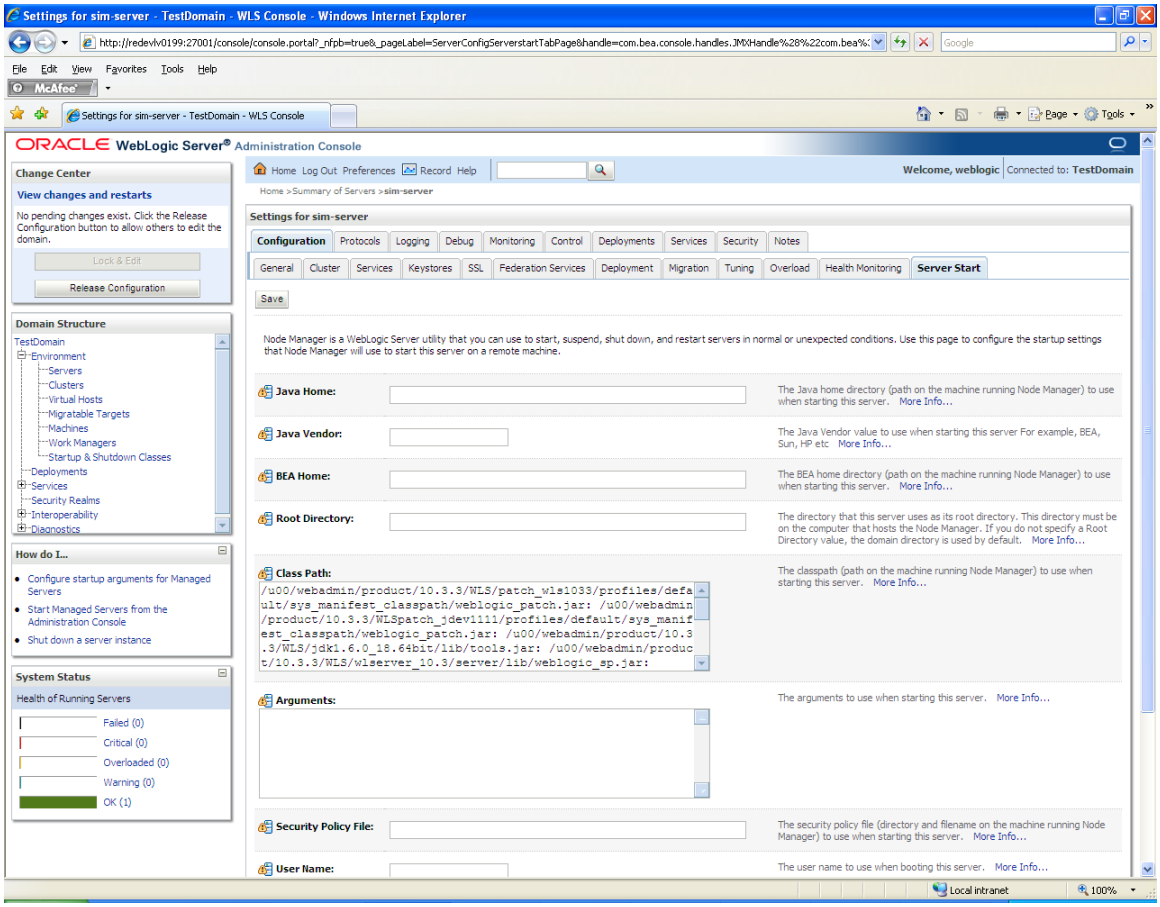
1. Navigate to Environments > Servers. Select sim-server. Click the Control tab.



2. Update in WebLogic console > servers > <app>-server->server start tab > Classpath with the following:

   CLASSPATH:

   <WLS_HOME>/patch_wls1033/profiles/default/sys_manifest_classpath/weblogic_patch.jar:
   <WLS_HOME>/patch_jdev1111/profiles/default/sys_manifest_classpath/weblogic_patch.jar: **<WLS_HOME>**/jdk1.6.0_18.64bit/lib/tools.jar:
   <WLS_HOME>/wlserver_10.3/server/lib/weblogic_sp.jar:
   <WLS_HOME>/wlserver_10.3/server/lib/weblogic.jar:
   <WLS_HOME>/modules/features/weblogic.server.modules_10.3.3.0.jar:
   <WLS_HOME>/wlserver_10.3/server/lib/webservices.jar:
   <WLS_HOME>/modules/org.apache.ant_1.7.1/lib/ant-all.jar:
   <WLS_HOME>/modules/net.sf.antcontrib_1.1.0.0_1-0b2/lib/ant-contrib.jar:
   <WLS_HOME>/modules/com.bea.core.apache.log4j_1.2.13.jar



3. Export WEBLOGIC_DOMAIN_HOME=<WLS_HOME>/user_projects/domains/<domain name>
4. Update <WLS_HOME>/server/lib/weblogic.policy file with the information below.

> **Note:** If copying the following text from this guide to UNIX, ensure that it is properly formatted in UNIX. Each line entry beginning with "permission" must terminate on the same line with a semicolon.

Note: <WEBLOGIC_DOMAIN_HOME> in the example below is the full path of the WebLogic domain; <managed_server> is the RPM managed server created. See the example. There should not be any space between file:<WEBLOGIC_DOMAIN_HOME.

```
grant codeBase
"file:<WEBLOGIC_DOMAIN_HOME>/servers/<managed_server>/tmp/_WL_user/-"
{
permission java.security.AllPermission;
permission
oracle.security.jps.service.credstore.CredentialAccessPermission "
credstoressp.credstore", "read,write,update,delete";
permission
oracle.security.jps.service.credstore.CredentialAccessPermission "
credstoressp.credstore.*", "read,write,update,delete";
};
grant codeBase
"file:<WEBLOGIC_DOMAIN_HOME>/servers/<managed_server>/cache/EJBCompil
erCache/-" {
permission java.security.AllPermission;
permission
oracle.security.jps.service.credstore.CredentialAccessPermission
"credstoressp.credstore", "read,write,update,delete";
permission
oracle.security.jps.service.credstore.CredentialAccessPermission
"credstoressp.credstore.*", "read,write,update,delete";
};

An example of the full entry that might be entered is:
grant codeBase
"file:/u00/webadmin/product/10.3.3/WLS/user_projects/domains/APPDomai
n/servers/sim-server/tmp/_WL_user/-" {
permission java.security.AllPermission;
permission
oracle.security.jps.service.credstore.CredentialAccessPermission "
credstoressp.credstore", "read,write,update,delete";
permission
oracle.security.jps.service.credstore.CredentialAccessPermission "
credstoressp.credstore.*", "read,write,update,delete";
};
grant codeBase
"file:/u00/webadmin/product/10.3.3/WLS/user_projects/domains/APPDomai
n/servers/sim-server/cache/EJBCompilerCache/-" {
permission java.security.AllPermission;
permission
oracle.security.jps.service.credstore.CredentialAccessPermission
"credstoressp.credstore", "read,write,update,delete";
permission
oracle.security.jps.service.credstore.CredentialAccessPermission
"credstoressp.credstore.*", "read,write,update,delete";
};
```

Set the LANG Environment Variable

The LANG environment variable must be set in the profile of the UNIX user who owns the application server ORACLE_HOME files. If you change the value of LANG or set the value for the first time, you must restart the Application Server in order for the change to take effect.

Example:

LANG=en_US

export LANG

Expand the SIM Application Distribution

To expand the SIM application distribution, do the following.

1. Log into the UNIX server as the user who owns the Web Logic installation. Create a new staging directory for the SIM application distribution (sim13application.zip). There should be a minimum of 300 MB disk space available for the application installation files.

   This location is referred to as INSTALL_DIR for the remainder of this chapter.
2. Copy sim13application.zip to <INSTALL_DIR> and extract its contents.

Run the SIM Application Installer

This installer configures and deploys the SIM application and Java WebStart client files.

1. Set the ORACLE_HOME, JAVA_HOME, and WEBLOGIC_DOMAIN_HOME environment variables. ORACLE_HOME should point to your WebLogic installation. JAVA_HOME should point to a valid Java 1.6 installation that is being used by WebLogic Application server. WEBLOGIC_DOMAIN_HOME should point to the full path of the domain into which SIM will be installed.
2. If you are using an X server such as Exceed, set the DISPLAY environment variable so that you can run the installer in GUI mode (recommended). If you are not using an X server, or the GUI is too slow over your network, unset DISPLAY for text mode.
3. Verify that the managed servers to which SIM will be installed are currently running.
4. Run the install.sh script. This launches the installer. After installation is completed, a detailed installation log file is created: <INSTALL_DIR>/sim/application/logs/sim-install-app.<timestamp>.log.

Note: The manual install option in the installer is not functional for this release. See the section, "Files not available to copy at the end of installation, results in non working applications – Weblogic only" in Appendix E: Common Installation Errors.

Note: See Appendix: SIM Application WebLogic Server Installer Screens for details on every screen and field in the WebLogic application installer.

Note: See Appendix: Common Installation Errors for details on common installation errors.

Clustered Installations – Post-Installation Steps

Skip this section if you are not clustering the application server.

If you are installing the SIM application to a clustered Web Logic Server environment, there are some extra steps you need to take to complete the installation. In these instructions, the application server node with the ORACLE_HOME you used for the SIM installer is referred to as the *master server*. All other nodes are referred to as the *remote server*.

Copy the <weblogic domain path>/<deployed sim client app name> directory from the master server to each remote server that is a member of the cluster that contains the deployed sim application.

Review and/or Configure Oracle Single Sign-On

Skip this section if you are not using Single Sign-On for user identification and authentication.

Single Sign-On is applicable only to the JnlpLaunch Servlet. The JnlpLaunch Servlet is a dynamically protected application. The JnlpLaunch Servlet causes the SIM client application to execute under the SSO user name with a temporary password.

Note: The JnlpLaunch servlet may be configured for either an SSO or non-SSO environment.

HTTP Server configuration requirements: The HTTP Server must be registered with the Oracle Single Sign-On server and the mod_osso module enabled. The registration process typically involves running the ssoreg.sh script at the OSSO server installation and copying the output osso.conf file to the HTTP Server. This process is documented in the Oracle Single Sign-On administration documentation.

JnlpLaunch requirements: The JnlpLaunch Servlet uses the configuration file, JnlpLaunch.properties, to control its behavior. Due to security considerations, this file must not be published or readable to the general public.

JnlpLaunch.properties has the following configuration entries that apply to Single Sign-On:

- `secret.key – Used` to create the temporary password, this property should contain a random string. If JnlpLaunch is deployed in a different JVM than the SIM Server EJBs, this string must be an exact match between the JnlpLaunch Servlet and the one available to the SIM EJBs. For security purposes, each separate instance of the SIM application (for example, test versus development) should have a different secret key.
- `user.validation.timeout` – Number of seconds the SIM Server uses to determine if a temporary password is still valid.
- `osso.used` – Determines if the JnlpLaunch Servlet will throw a 499 error when an unathenticated user has been detected. This property must be set to True if Oracle Single Sign-On is used and False if not.

The JnlpLaunch.properties file is initialized by the SIM installer and should contain valid entries for SSO when the Enable Single Sign-On in SIM? prompt was answered by a Y or Yes. However, an administrator may want to alter the user.validation.timeout or other property after the initial installation.

For instructions for setting up SSO for WebLogic, see Appendix: Oracle Single Sign-On for WebLogic.

SIM Batch Scripts

The SIM batch programs are installed in the location that was specified during application installation.

The batch programs can be run from a different location if you cannot run them from under the application server <WEBLOGIC_DOMAIN_HOME>.. To install the batch files in a different location just copy the entire batch folder to the appropriate destination.

The batch directory is assumed to be located on the same server as the application server. If you copy the SIM batch directory to a location on a different server, then you need to configure the file path to the sim-batch.log file, which is defined in batch/resources/log4j.xml.

See the "Batch Detail" section of the *Oracle Retail Store Inventory Management Operations Guide* for information about how to run batches.

Configuring Web Service Security in SIM

SIM web service is pre-configured with username-token-digest security policy. This can be verified by looking at the wsdl:

```
<s0:definitions targetNamespace="http://retail.oracle.com/SIM/sim-ws.wsdl"
xmlns="" xmlns:s0="http://schemas.xmlsoap.org/wsdl/"
xmlns:s1="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:s10="http://retail.oracle.com/SIM/InventoryAdjustment/v1"
xmlns:s11="http://retail.oracle.com/SIM/Transfer/v1"
xmlns:s12="http://retail.oracle.com/SIM/StockReturn/v1"
xmlns:s13="http://retail.oracle.com/SIM/ItemImage/v1"
xmlns:s14="http://retail.oracle.com/SIM/SIMBaseType/v1"
xmlns:s15="http://retail.oracle.com/SIM/sim-ws.wsdl"
xmlns:s16="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:s2="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
xmlns:s3="http://retail.oracle.com/SIM/ItemLookup/v1"
xmlns:s4="http://retail.oracle.com/SIM/InventoryLookup/v1"
xmlns:s5="http://retail.oracle.com/SIM/UINSerialization/v1"
xmlns:s6="http://retail.oracle.com/SIM/CustomerOrder/v1"
xmlns:s7="http://retail.oracle.com/SIM/ItemBasket/v1"
xmlns:s8="http://retail.oracle.com/SIM/PosTransaction/v1"
xmlns:s9="http://retail.oracle.com/SIM/SaleReturnTxn/v1"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
  <s0:documentation>This WSDL document describes the Oracle Retail Store Inventory
Management(SIM) service.</s0:documentation>
  <s1:Policy s2:Id="ws-policy">
    <sp:SupportingTokens xmlns:sp="http://docs.oasis-open.org/ws-sx/ws-
securitypolicy/200702">
      <wsp:Policy>
        <sp:UsernameToken sp:IncludeToken="http://docs.oasis-open.org/ws-sx/ws-
securitypolicy/200702/IncludeToken/AlwaysToRecipient">
          <wsp:Policy>
            <sp:HashPassword/>
            <sp:WssUsernameToken10/>
          </wsp:Policy>
        </sp:UsernameToken>
```

```
      </wsp:Policy>
    </sp:SupportingTokens>
  </s1:Policy>
  <wsp:UsingPolicy s0:Required="true"/>
```

User and role have to be configured through WebLogic admin console and can be done as follows:

1. Click the **Security Realms** link in the **Domain Structure** window.
2. Default realm is displayed. Click on the link for the realm.
3. Click the **Users and Groups** tab.
4. Click **New**. Enter user name (for example, simwsuser) and password on the next screen. Leave the default value for Provider.
5. Click **OK** to save the changes. It will show the new user in the list of users
6. Navigate to the **Security** tab of SIM web service.
7. Click **New** in the Roles tab.
8. Enter the role name in the **Name** field (for example, simwsrole). Leave the **Provider Name** as default.
9. Click **OK**. It will show the newly created role in the role tab.
10. Now we will add the user to the role. Click the newly created role.
11. Click **Add Conditions**.
12. Select **User** in the **Predicate List** drop-down and click **Next**.
13. Enter the user name that was created in the security realm and click **Add**. It will get added to the list below the text box.
14. Click **Finish**.
15. Click **Save**.
16. Navigate back to the **Security > Policies** tab of the Web service.
17. Click **Add Conditions**.
18. Select **Role** in the **Predicate List** drop down. Click **Next**.
19. Enter the role name that was created earlier. Click **Add**. The role is added to the list below the text box.
20. Click **Finish**.
21. Click **Save**.

Resolving Errors Encountered During Application Installation

If the application installer encounters any errors, it halts execution immediately. You can run the installer in silent mode so that you do not have to retype the settings for your environment. See Appendix D of this document for instructions on silent mode.

See Appendix E of this document for a list of common installation errors.

Since the application installation is a full reinstall every time, any previous partial installs are overwritten by the successful installation.

Web Help Files

The application installer automatically copies the web help files to the proper location. They are accessible from the help links within the application.

Starting and Stopping the Wavelink Server

In order to use handheld wireless devices with SIM, the Wavelink server must be running. The SIM application installer installs, configures, and starts the Wavelink server for you, so once the SIM application install is complete, the Wavelink server is ready to be used.

Note: If you use the Enterprise Manager to restart SIM, then you need to restart the Wavelink server manually.

If you use SIM's startup and shutdown scripts to restart SIM on the command line, then the Wavelink server will also be restarted along with SIM. However, if you use the Enterprise Manager to restart SIM, the Wavelink server is not affected. Thus it needs to be restarted manually once SIM is running again.

The Wavelink server scripts are installed into the <sim-wireless-directory>/bin.

Note: The wireless functionality in SIM is dependent on Wavelink and includes a client and server component. Wavelink software ensures that the wireless user interface of SIM can work with various handheld devices.

For the handheld to interact correctly with SIM, it is required to install the appropriate Wavelink studio client. The Wavelink studio client and its installation instructions can be found at http://www.wavelink.com/download/downloads.aspx.

The Oracle Retail Wireless Foundation Server is bundled with the SIM server. It has a single session free license. For multiple sessions additional licenses need to be obtained.

Contact your Oracle sales representative or client partner for Wavelink Studio Client and Oracle Retail Wireless Foundation Server license information.

Note: For configurations of physical handheld devices or wireless network setup, check your hardware manufacturer's manual or Wavelink's studio client information. This information is not covered in this guide.

7

Data Seeding

After SIM database and application completes, the foundation data must be data seeded into SIM before a successful login can occur. Foundational data includes store, item, supplier, warehouse, and merchandise hierarchy (for example, department, class, and subclass).

SIM provides data seeding utility to import foundation and store data from Oracle Retail Merchanding System (ORMS) into SIM.

For data seeding details, see the Data Seeding section of the *Oracle Retail Store Inventory Management Implementation Guide, Volume 1*.

Data seed from RMS13.2 database, unzip the sim-database-full.zip, run data seeding from sim/dbschema/dbscripts/data_seeding/RMS_13_2 folder.

(If dataseeding from RMS13.1 database, run data seeding from sim/dbschema/dbscripts/data_seeding/RMS_13_1 folder)

This folder is referred to as DATA_SEEDING_DIR for the remainder of this chapter.

> **Note:** The SIDs for RMS and SIM databases should exist in the machine where these data seeding scripts are run.

1. Set the following environment variables:
   - Set ORACLE_SID to the name of SIM database.

     For example:

     ```
     export ORACLE_SID=<SIM_DB_NAME>
     ```
   - Set the ORACLE_HOME. It is recommended that this be the ORACLE_HOME of the SIM database.

     For example:

     ```
     export  ORACLE_HOME=/u00/oracle/product/11.2.0.1
     ```
   - Set the PATH to: $ORACLE_HOME/bin: (SQLLOADER and SQLPLUS utilities are used for data seeding which are available in this path)

     ```
     export PATH=$ORACLE_HOME/bin:$PATH
     ```
2. Change to the <DATA_SEEDING_DIR> directory. Verify the directory and the file permissions. The recommended permissions for data seeding directories or files are 775 (rwxrwxr-x).
3. Set the environment variables for the following:
   - export RMS_DB=<RMS_DB>
   - export RMS_USER=<RMS_USER>
   - export RMS_PWD=<RMS_PWD>
   - export SIM_DB=<SIM_DB>
   - export SIM_USER=<SIM_USER >
   - export SIM_PWD=<SIM_PWD)>
4. Modify data_seedingSet values for following variables if non-default values are desired:
   - SIM_COMMIT_BLOCK=<SIM_COMMIT_BLOCK > (default to 1000)

> **Note:** The default shell used in data seeding scripts is '#!/usr/bin/env bash . For data seeding run in other platform other than linux, if encounter a syntax error due to the platform dependency, by modifying the shell to other shell (e.g ksh93) may resolve the platform depency issue.

5. Run the data_seeding.sh command under the <DATA_SEEDING_DIR> directory:
   `./data_seeding.sh`

   The data_seeding process prompts the user to choose the execution option. The sequence of the data seeding is as follows:

   a. Foundation Data Seeding
   b. Data Seeding Foundation Verification
   c. Store Data Seeding

      After choosing the Store Data Seeding option, you are prompted with the following message. "Press Enter for all stores [OR] Enter comma (,) separated Store ID's with NO SPACE in between." The options are:

      – All stores: Press Enter to import data for ALL stores.
      – List of Stores: Enter store IDs separated by a comma, and then press Enter.

   d. Data Seeding Store Verification
   e. Data Seeding Clean up
   f. UIN Attributes Setup

      > **Note:** For stores that use items with unique identification numbers (UIN), the UIN Attributes setup inserts/updates the UIN attributes on depart/class level whenever a new store is created. See the UpdateUinAttributes Batch section of the *Oracle Retail Store Inventory Management Operations Guide* for details.

6. Check the SIM schema for any disabled constraints.

   > **Example:** SELECT * from DBA_CONSTRAINTS where OWNER='SIMUSER' and STATUS='DISABLED'

The above query should return no rows. If there are any disabled constraints, you should enable them at this time.

> **Example:** ALTER TABLE <TABLE> enable constraint <CONSTRAINT>

> **Post-Data-Seeding-Clean-up:** After data seeding is finished, and you are convinced that your data was correctly seeded, you can remove dat, dsc, log and bad files.

8

Test the SIM Application

Once SIM database and application are installed, foundation data is imported into SIM, you should have a working SIM application installation. To launch the application client, open a web browser and go to the client URL. You can find the URL in the next steps section of the log file that was produced by the installer.

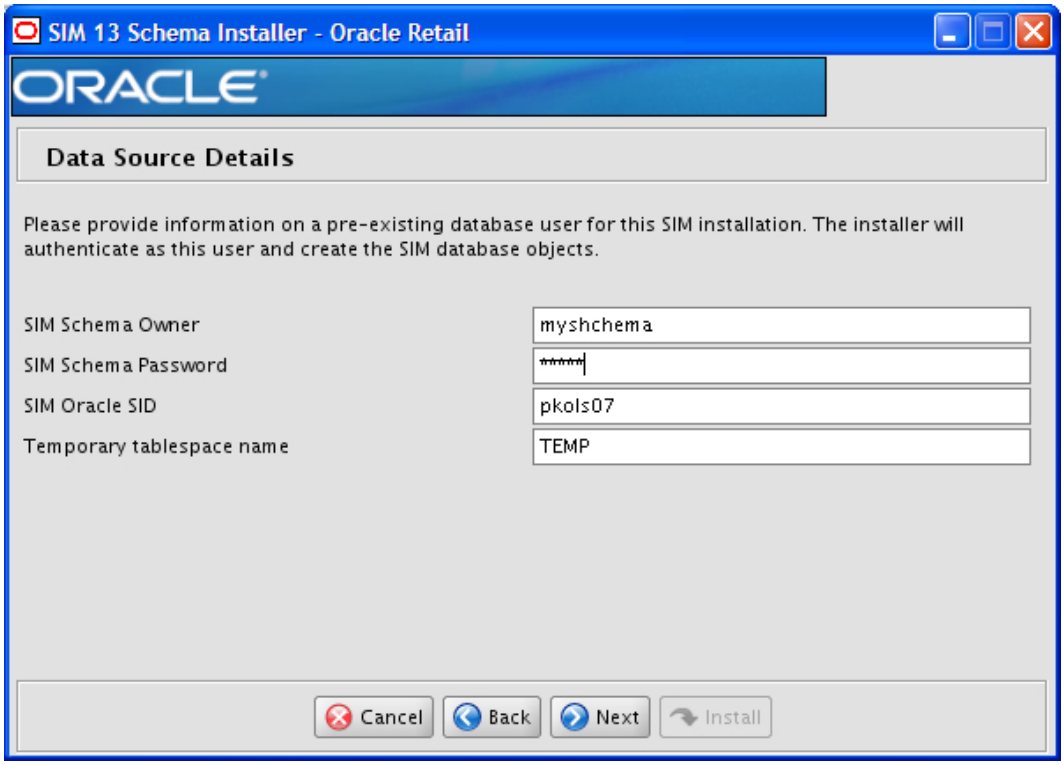
Example: http://myhost:7777/sim-client/launch?template=sim_jnlp_template.vm

A

Appendix: SIM Database Schema Installer Screens

You need the following details about your environment for the installer to successfully install the SIM database schema. Depending on the options you select, you may not see some screens.

Screen: Data Source Details



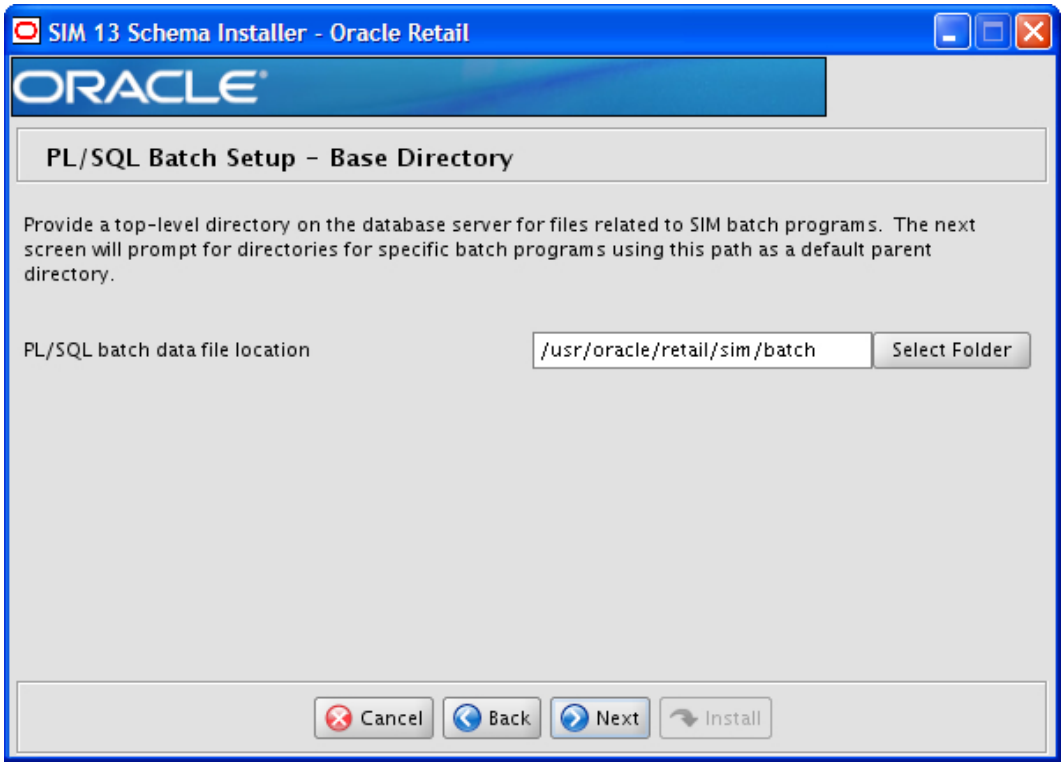
Field Title	SIM Schema Owner
Field Description	The pre-existing database user for this installation
Destination	sim_dba.sql, dataseeding.cfg
Examples	myschema

Field Title	Sim Schema Password
Field Description	The SIM Schema Owner's password

Field Title	SIM Oracle SID
Field Description	The name of the database where the SIM schema will be installed
Examples	mydb

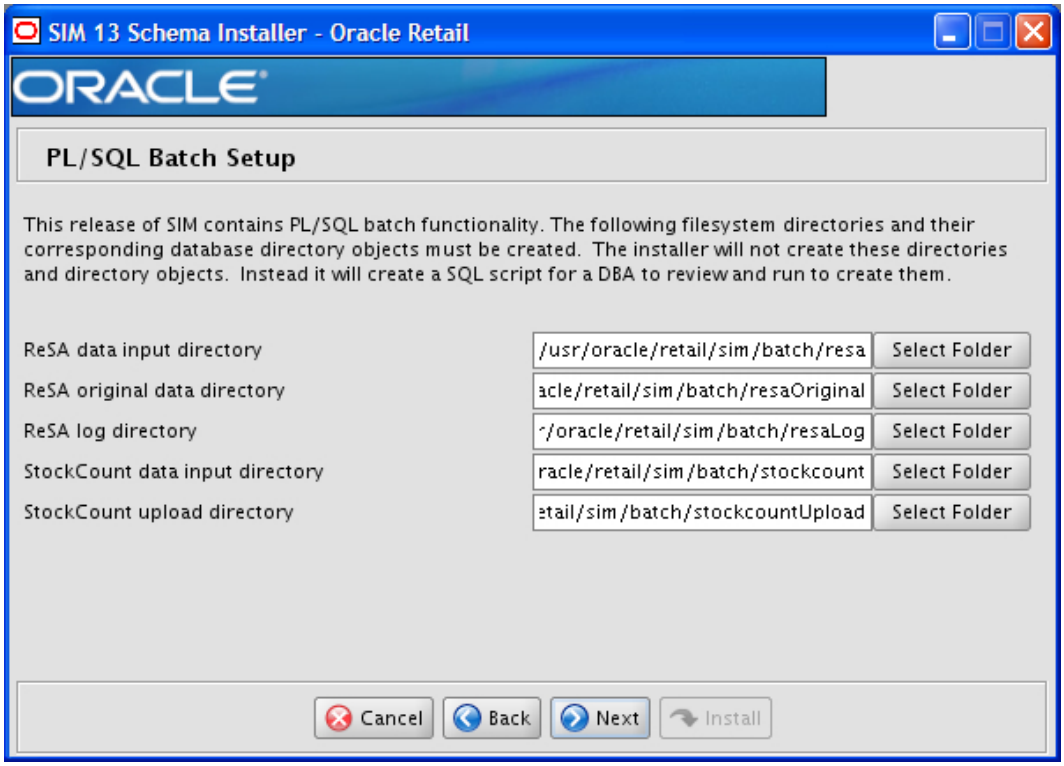
Field Title	Temporary tablespace name
Field Description	Temporary tablespace provided to the create_user.sql script at the time that the SIM database user was created.
Examples	TEMP

Screen: PL/SQL Batch Setup – Base Directory



Field Title	PL/SQL batch data file location
Field Description	A directory which will be the parent directory for all other PL/SQL batch processing directories. **Note:** The user must have access permissions for this directory.
Destination	sim_dba.sql
Examples	/usr/oracle/retail/sim/batch

Screen: PL/SQL Batch Setup (3 screens)



Field Title	ReSA data input directory
Field Description	A filesystem directory and database directory object used for processing ReSA data
Destination	sim_dba.sql
Examples	/usr/oracle/retail/sim/batch/resa
Notes	The installer will not create these directories or directory objects. It will produce the sim_dba.sql script, which can be used to create them.

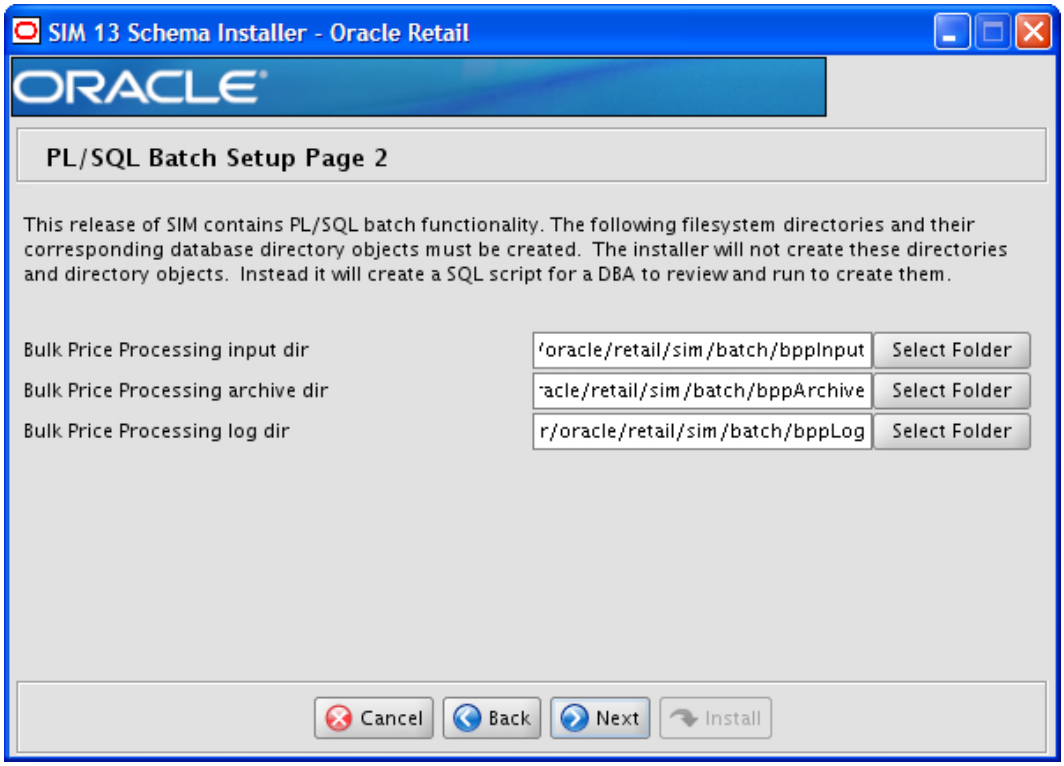
Field Title	ReSA original data directory
Field Description	A filesystem directory and database directory object used for processing ReSA data
Destination	sim_dba.sql
Examples	/usr/oracle/retail/sim/batch/resa
Notes	The installer will not create these directories or directory objects. It will produce the sim_dba.sql script, which can be used to create them.

Field Title	ReSA log directory
Field Description	A filesystem directory and database directory object used to place logs of ReSA data
Destination	sim_dba.sql
Examples	/usr/oracle/retail/sim/batch/resaLog

Field Title	StockCount data input directory
Field Description	A filesystem directory and database directory object used to process StockCount data
Destination	sim_dba.sql
Examples	/usr/oracle/retail/sim/batch/stockcount
Notes	The installer will not create these directories or directory objects. It will produce the sim_dba.sql script, which can be used to create them.

Field Title	StockCount upload directory
Field Description	A filesystem directory and database directory object used to process StockCount data
Destination	sim_dba.sql
Example	/usr/oracle/retail/sim/batch/stockcountUpload
Notes	The installer will not create these directories or directory objects. It will produce the sim_dba.sql script, which can be used to create them.

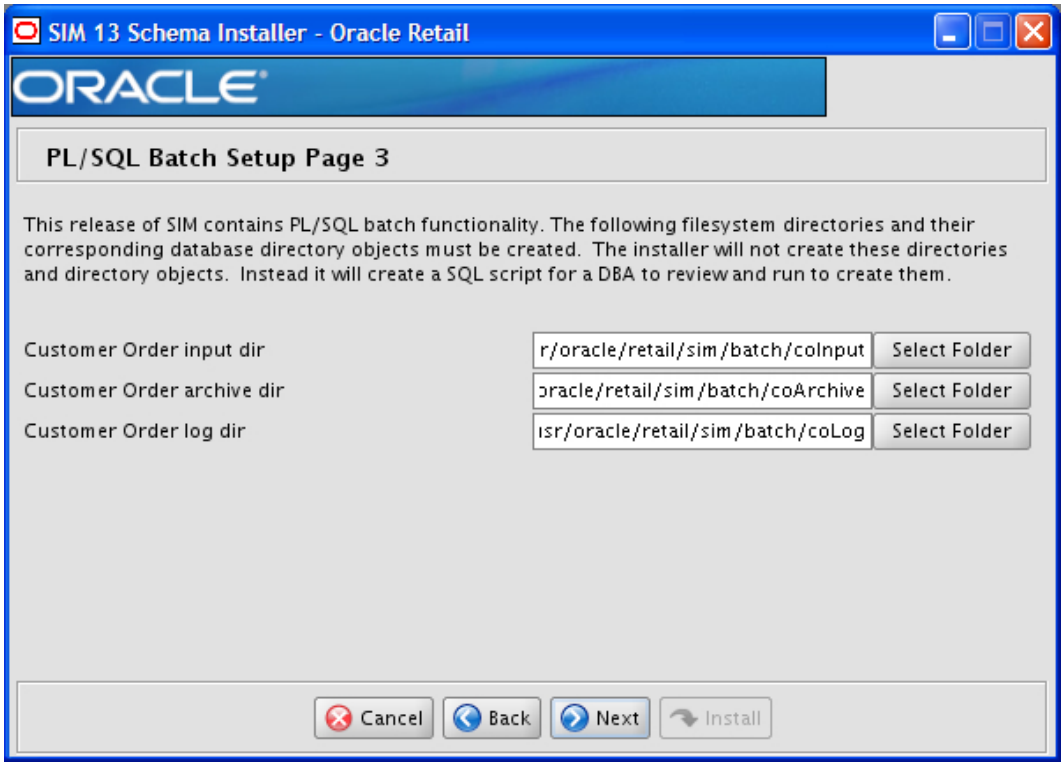
Screen: PL/SQL Batch Setup Page 2 (3 screens)



Field Title	Bulk Price Processing input dir
Field Description	A filesystem directory and database directory object used for placing the input price change (Promotion, Clearance and Regular Price Change) files. The SQL procedure, reads the file from this input file from this directory.
Destination	sim_dba.sql
Example	/usr/oracle/retail/sim/batch/bppInput

Field Title	Bulk Price Processing log dir
Field Description	A filesystem directory and database directory object used to place logs of bulk price processing execution details
Destination	sim_dba.sql
Example	/usr/oracle/retail/sim/batch/bppLog

Screen: PL/SQL Batch Setup Page 3 (3 screens)



Field Title	Customer Order input dir
Field Description	A filesystem directory and database directory object used for placing the input customer order files. The SQL procedure, reads the file from this input file from this directory.
Destination	sim_dba.sql
Example	/usr/oracle/retail/sim/batch/coInput

Field Title	Customer Order archive dir
Field Description	A filesystem directory and database directory object used to place the fully processed customer order procedure
Destination	sim_dba.sql
Example	/usr/oracle/retail/sim/batch/coArchive

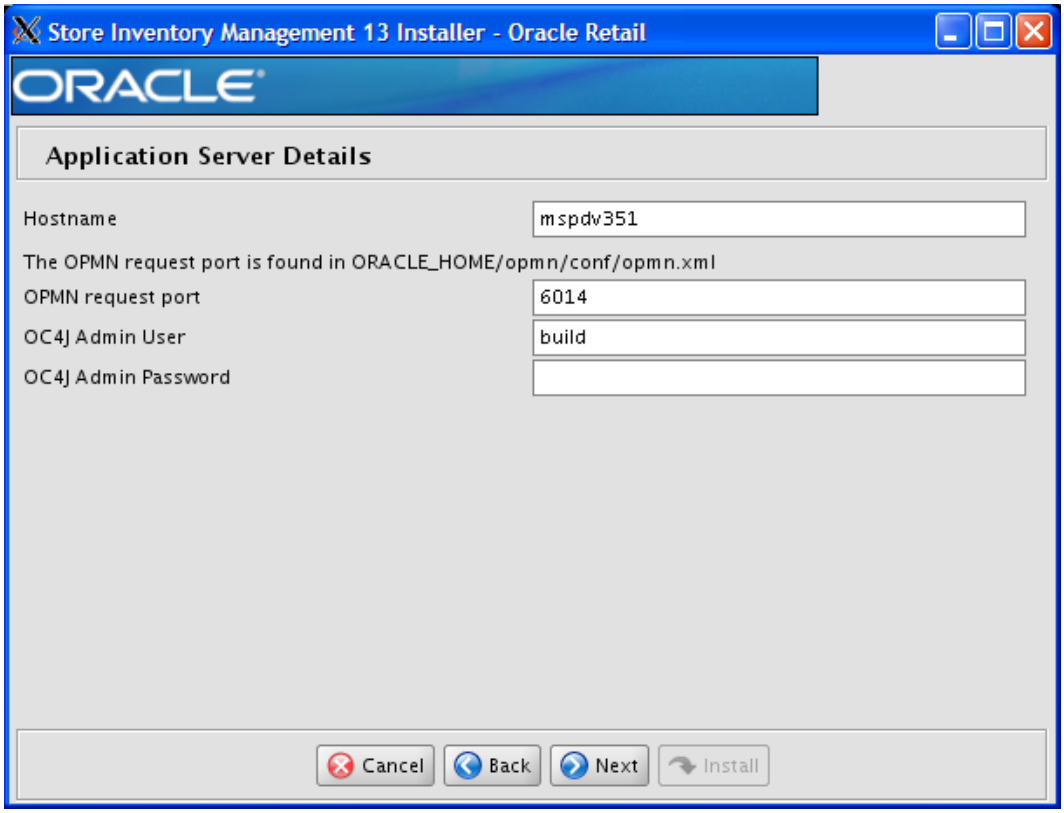
Field Title	Customer Order log dir
Field Description	A filesystem directory and database directory object used to place logs of customer order processing execution details
Destination	sim_dba.sql
Example	/usr/oracle/retail/sim/batch/coLog

B

Appendix: SIM Application Oracle Application Server (OAS) Installer Screens

You need the following details about your environment for the installer to successfully deploy the SIM application. Depending on the options you select, you may not see some screens.

Screen: Application Server Details



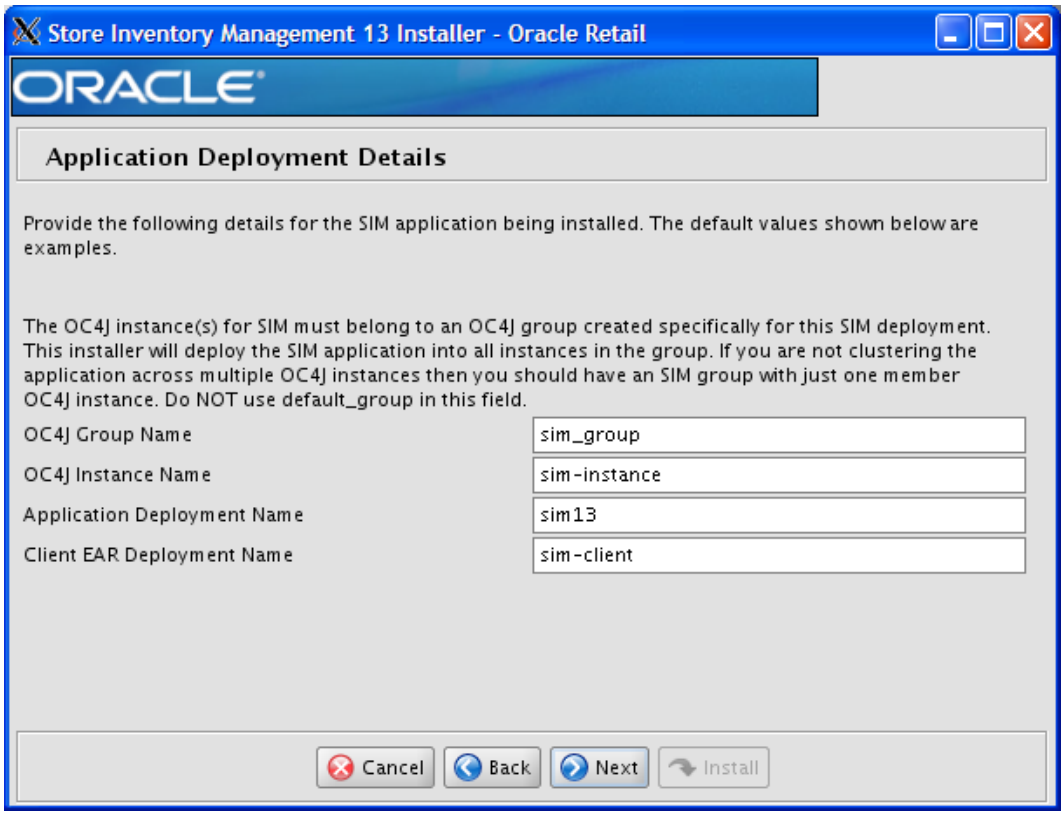
Field Title	Hostname
Field Description	The hostname of the server where the application server is installed
Destination	client.cfg, jndi.cfg,JnlpLaunch.properties
Example	myhost.us.oracle.com
Notes	Used by installer scripts to deploy EAR and WAR files and to create default inputs for client codebase and JNDI provider URL

Field Title	OPMN request port
Field Description	The OPMN request port found in $ORACLE_HOME/opmn/conf/opmn.xml <port local="6100" remote="6200" request="6003"/>
Example	6003
Notes	Used by installer scripts to deploy EAR and WAR files and to create default input for JNDI provider URL

Field Title	OC4J Admin User
Destination	jndi.cfg, JnlpLaunch.properties
Example	oc4jadmin
Notes	Used by installer scripts to deploy EAR and WAR files

Field Title	OC4J Admin Password
Field Description	The password of the OC4J Admin User
Destination	jndi.cfg, JnlpLaunch.properties
Notes	Used by installer scripts to deploy EAR and WAR files

Screen: Application Deployment Details



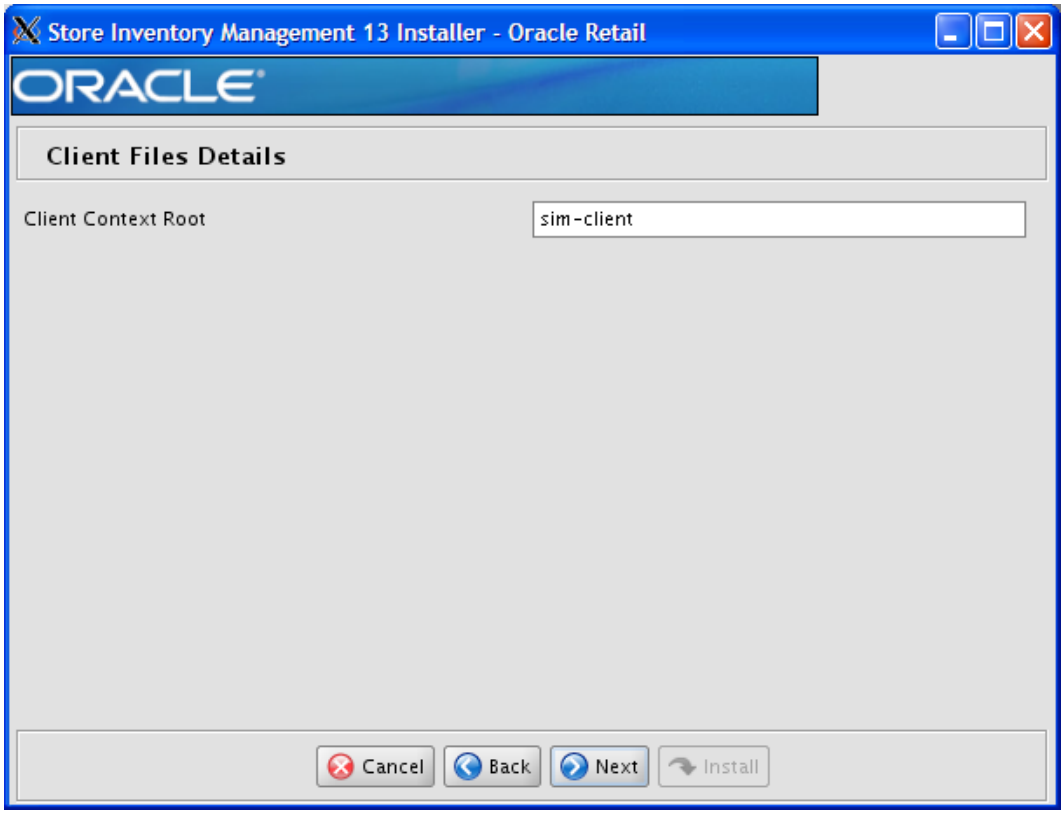
Field Title	OC4J Group Name
Field Description	Name of the OC4J group that was created for this SIM application. The OC4J instance given in the OC4J Instance Name field should be a member of this group. The installer deploys the SIM application to all OC4J instances which are members of this group. For this reason, you should not use default_group. A new group dedicated to SIM should be created instead.
Example	sim_group

Field Title	OC4J Instance Name
Field Description	The name of the OC4J instance that the SIM application will be deployed to
Destination	log4j.xml, MANIFEST.MF, startup.sh, shutdown.sh
Example	sim-oc4j-instance

Field Title	Application Deployment Name
Field Description	The name that will be used by the application server to identify the SIM application
Example	sim13
Notes	Used by installer scripts to deploy the application and to create default values for JNDI provider URL

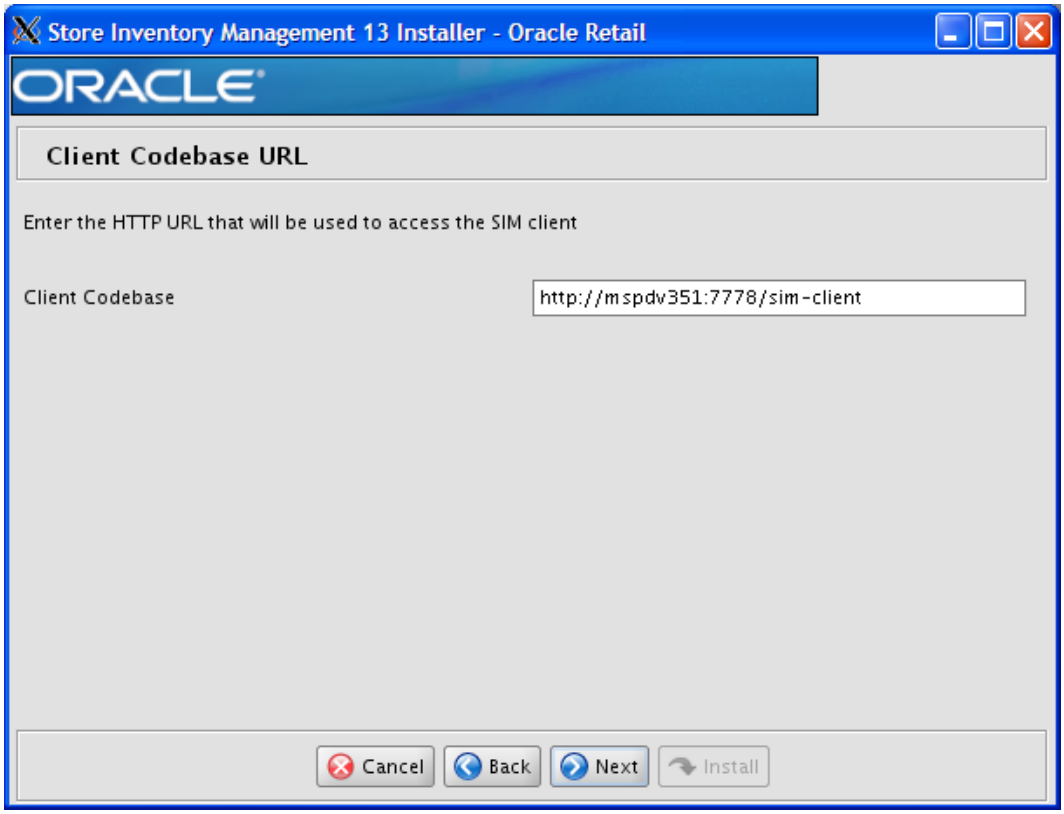
Field Title	Client EAR Deployment Name
Field Description	The name that will be used by the application server to deploy the sim-client.ear file.
Example	sim-client

Screen: Client Files Details



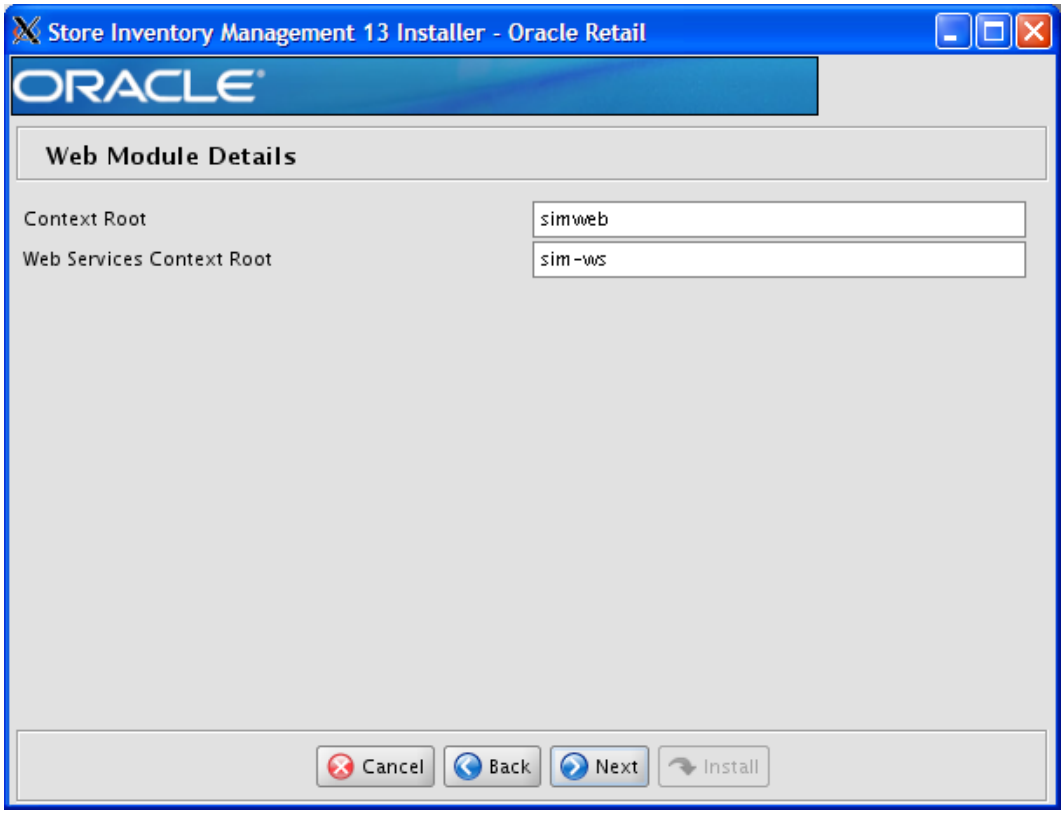
Field Title	Client Context Root
Field Description	Context root for sim-client.war
Destination	JnlpLaunch.properties
Example	sim-client
Notes	Used by installer to create default value for Client Codebase URL

Screen: Client Codebase URL



Field Title	Client Codebase
Field Description	The HTTP URL that points to the SIM client installation. The URL is made up of the Hostname, the HTTP port, and the Client Context Root.
Destination	JNLPLaunch.properties, sim_config.jnlp, client.cfg
Example	http://myhost:7777/sim-client
Notes	The Client Codebase URL must match the Client Context Root from the previous screen.

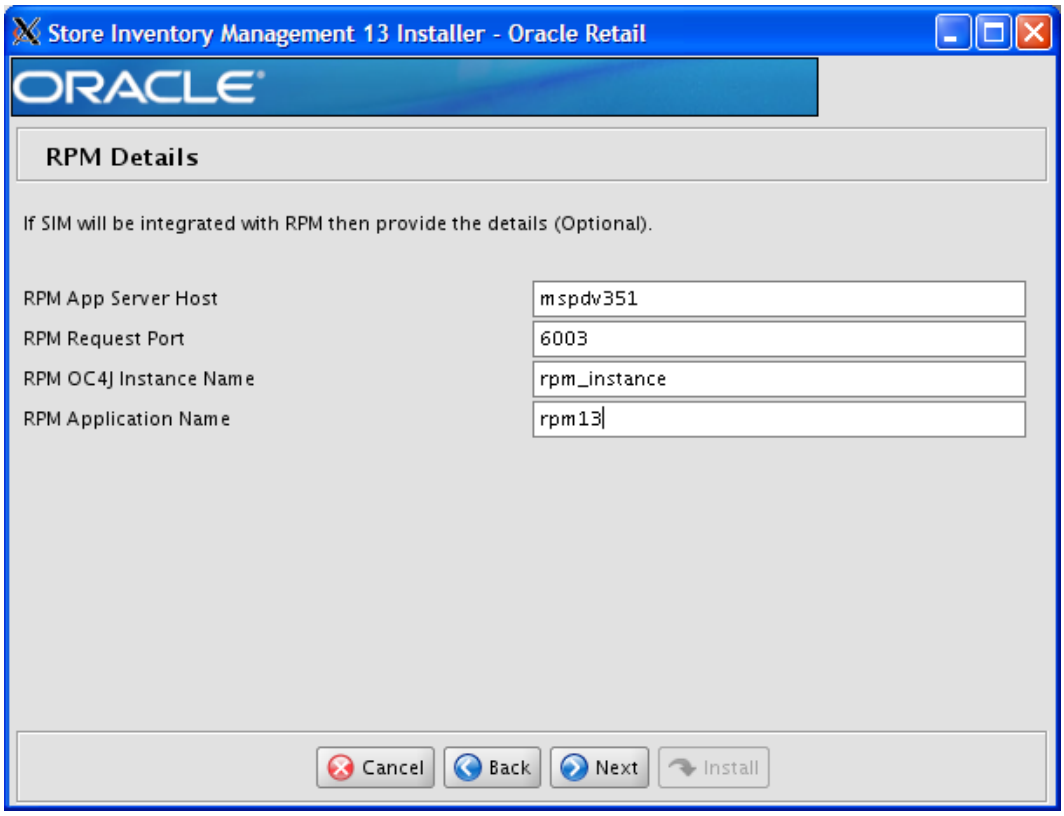
Screen: Web Module Details



Field Title	Context Root
Field Description	The context root for sim.war
Destination	application.xml
Example	simweb

Field Title	Web Services Context Root
Field Description	The context root for sim-ws.war
Destination	application.xml
Example	sim-ws

Screen: RPM Details



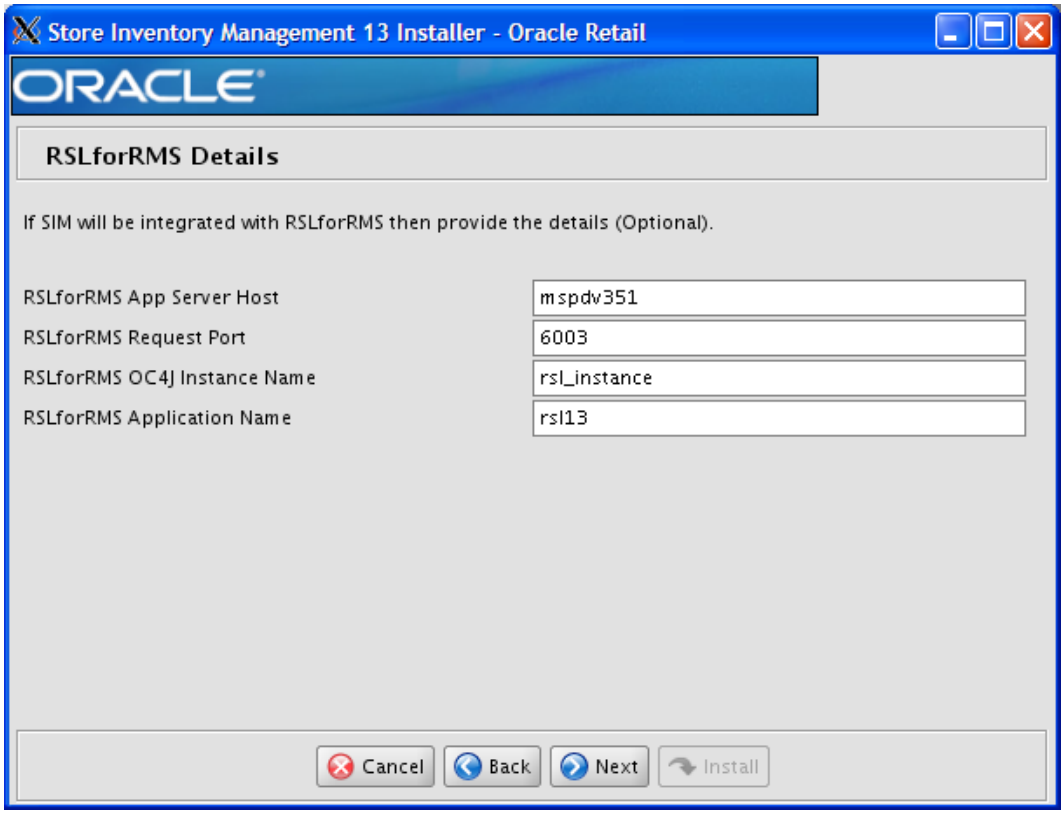
Field Title	RPM App Server Host
Field Description	The name of the application server host where the RPM application is installed
Destination	jndi_providers.xml
Example	myhost.us.oracle.com
Notes	Used only if integrating SIM with RPM

Field Title	RPM Request Port
Field Description	The OPMN request port for the application server where RPM is intalled. The OPMN request port is found in $ORACLE_HOME/opmn/conf/opmn.xml <port local="6100" remote="6200" request="6003"/>
Destination	jndi_providers.xml
Example	6003
Notes	Used only if integrating SIM with RPM

Field Title	RPM OC4J Instance Name
Field Description	The name of the OC4J instance where the RPM application is installed
Destination	jndi_providers.xml
Example	rpm-o4cj-instance
Notes	Used only if integrating SIM with RPM

Field Title	RPM Application Name
Field Description	The name that will be used by the application server to identify the RPM application
Destination	jndi_providers.xml
Example	rpm13
Notes	Used only if integrating SIM with RPM

Screen: RSLforRMS Details



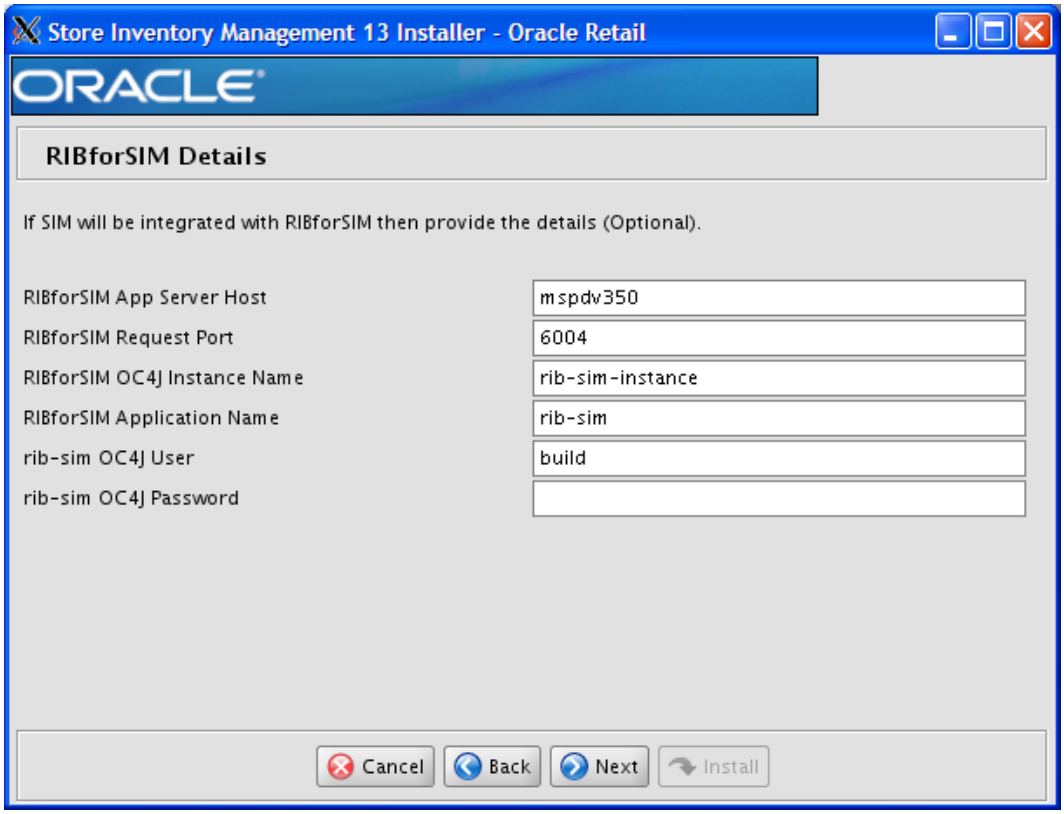
Field Title	RSLforRMS App Server Host
Field Description	The name of the application server host where the RSLforRMS application is installed
Destination	jndi_providers.xml
Example	myhost.us.oracle.com
Notes	Used only if integrating SIM with RSLforRMS

Field Title	RSLforRMS Request Port
Field Description	The OPMN request port for the application server where RSLforRMS is intalled. The OPMN request port is found in $ORACLE_HOME/opmn/conf/opmn.xml <port local="6100" remote="6200" request="6003"/>
Destination	jndi_providers.xml
Example	6003
Notes	Used only if integrating SIM with RSLforRMS

Field Title	RSLforRMS OC4J Instance Name
Field Description	The name of the OC4J instance where the RSLforRMS application is installed
Destination	jndi_providers.xml
Example	rsl-rms-o4cj-instance
Notes	Used only if integrating SIM with RSLforRMS

Field Title	RSLforRMS Application Name
Field Description	The name that will be used by the application server to identify the RSLforRMS application
Destination	jndi_providers.xml
Example	rsl-rsm
Notes	Used only if integrating SIM with RSLforRMS

Screen: RIBforSIM Details



Field Title	RIBforSIM App Server Host
Field Description	The name of the application server host where the RIBforSIM application is installed
Destination	jndi_providers_ribclient.xml
Example	myhost.us.oracle.com
Notes	Used only if integrating SIM with RIBforSIM

Field Title	RIBforSIM Request Port
Field Description	The OPMN request port for the application server where RIBforSIM is intalled. The OPMN request port is found in $ORACLE_HOME/opmn/conf/opmn.xml <port local="6100" remote="6200" request="6003"/>
Destination	jndi_providers_ribclient.xml
Example	6003
Notes	Used only if integrating SIM with RIBforSIM

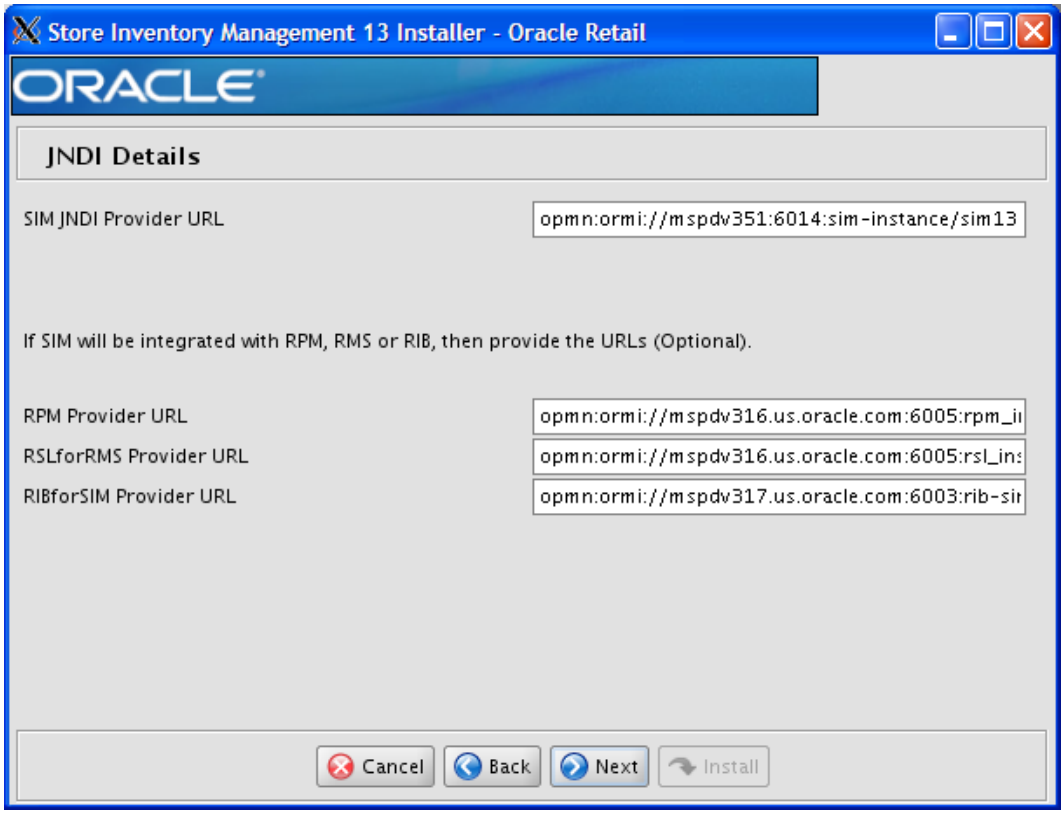
Field Title	RIBforSIM OC4J Instance Name
Field Description	The name of the OC4J instance where the RIBforSIM application is installed
Destination	jndi_providers_ribclient.xml
Example	rib-sim-o4cj-instance
Notes	Used only if integrating SIM with RIBforSIM

Field Title	RIBforSIM Application Name
Field Description	The name that will be used by the application server to identify the RIBforSIM application
Destination	jndi_providers_ribclient.xml
Example	rib-sim
Notes	Used only if integrating SIM with RIBforSIM

Field Title	rib-sim OC4J User
Field Description	The OC4J Admin User for the OC4J instance where rib-sim is installed.
Destination	jndi_providers_ribclient.xml
Example	oc4jadmin
Notes	Used only if integrating SIM with RIBforSIM

Field Title	rib-sim OC4J Password
Field Description	The password of the OC4J Admin User for the OC4J instance where rib-sim is installed.
Destination	jndi_providers_ribclient.xml
Notes	Used only if integrating SIM with RIBforSIM

Screen: JNDI Details



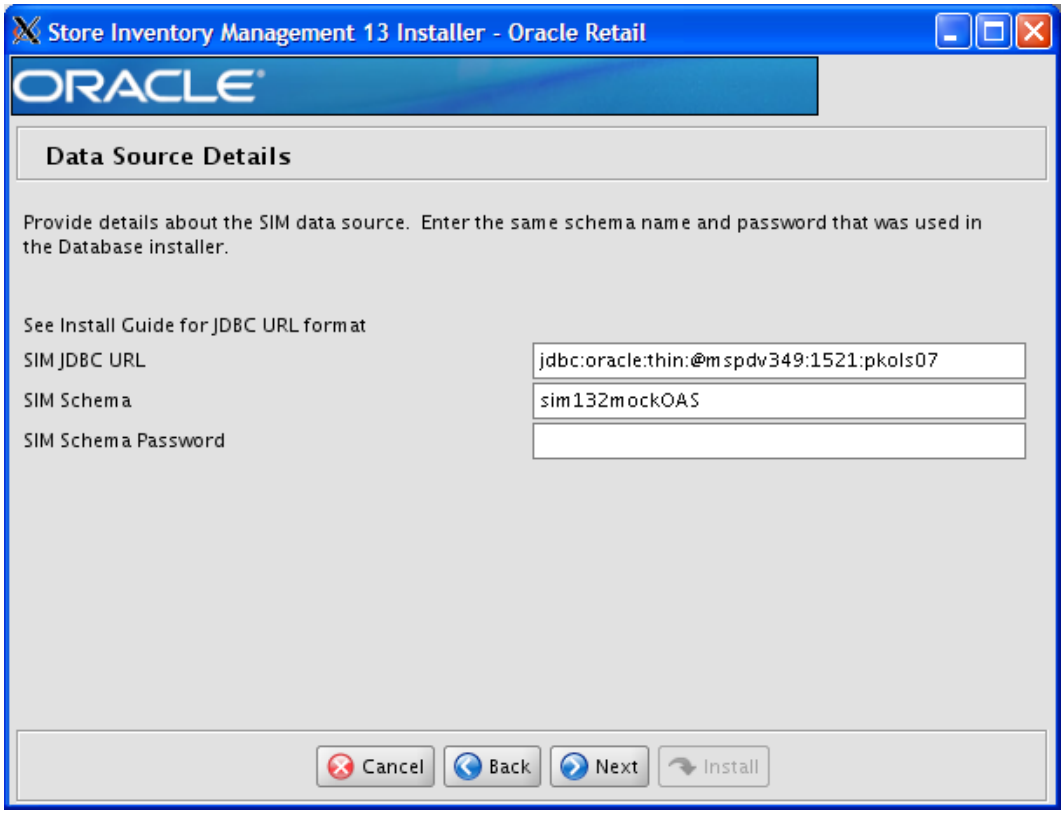
Field Title	SIM JNDI Provider URL
Field Description	JNDI provider URL for the SIM application
Destination	jndi.cfg, JnlpLaunch.properties
Example	opmn:ormi://myhost.us.oracle.com:6003:sim-oc4j-instance/sim13
Notes	Confirm the JNDI provider URL, which is constructed based on previous inputs for Hostname, OPMN Request Port, OC4J Instance Name, and Application Deployment Name

Field Title	RPM Provider URL
Field Description	JNDI provider URL for the RPM application
Destination	jndi_providers.xml
Example	opmn:ormi://myhost.us.oracle.com:6003:rpm-oc4j-instance/rpm13
Notes	Confirm the JNDI provider URL, which is constructed based on previous inputs for Hostname, OPMN Request Port, OC4J Instance Name, and Application Deployment Name

Field Title	RSLforRMS Provider URL
Field Description	JNDI provider URL for the RSLforRMS application
Destination	jndi_providers.xml
Example	opmn:ormi://myhost.us.oracle.com:6003:rsl-rms-oc4j-instance/rsl
Notes	Confirm the JNDI provider URL, which is constructed based on previous inputs for Hostname, OPMN Request Port, OC4J Instance Name, and Application Deployment Name

Field Title	RIBforSIM Provider URL
Field Description	JNDI provider URL for the RIBforSIM application
Destination	jndi_providers.xml
Example	opmn:ormi://myhost.us.oracle.com:6003:rib-sim-oc4j-instance/rib-sim
Notes	Confirm the JNDI provider URL, which is constructed based on previous inputs for Hostname, OPMN Request Port, OC4J Instance Name, and Application Deployment Name

Screen: Data Source Details

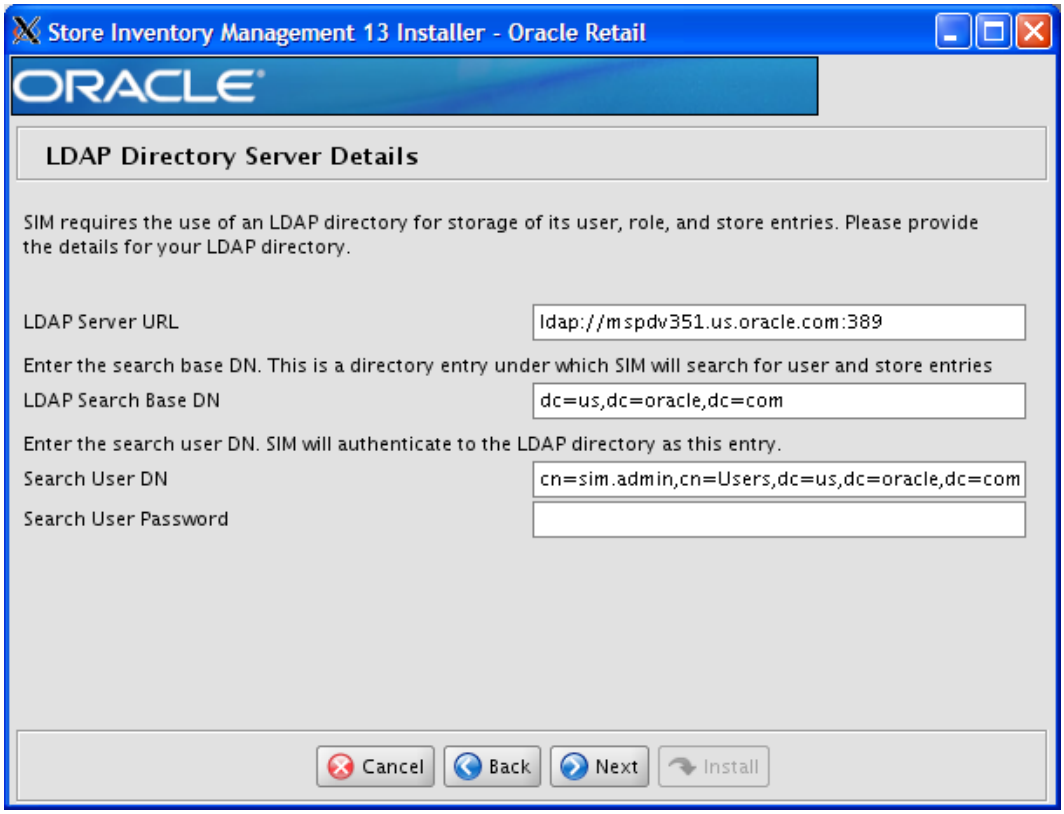


Field Title	SIM JDBC URL
Field Description	URL used by the SIM application to access the SIM database schema.
Destination	data-sources.xml
Example	jdbc:oracle:thin:@myhost:1521:mydatabase jdbc:oracle:thin:@(DESCRIPTION =(ADDRESS_LIST =(ADDRESS = (PROTOCOL = TCP)(HOST = myhost1)(PORT = 1521))(ADDRESS = (PROTOCOL = TCP)(HOST = myhost2)(PORT = 1521))(LOAD_BALANCE = yes))(CONNECT_DATA =(SERVICE_NAME = mydatabase)))

Field Title	SIM Schema
Field Description	The schema name
Destination	data-sources.xml
Notes	The schema name should match the name you provided when you ran the database schema installer.

Field Title	SIM Schema Password
Field Description	The password for the SIM Schema
Destination	data-sources.xml

Screen: LDAP Directory Server Details



Field Title	LDAP Server URL
Field Description	URL for your LDAP directory server.
Destination	ldap.cfg
Example	ldap://myhost:389

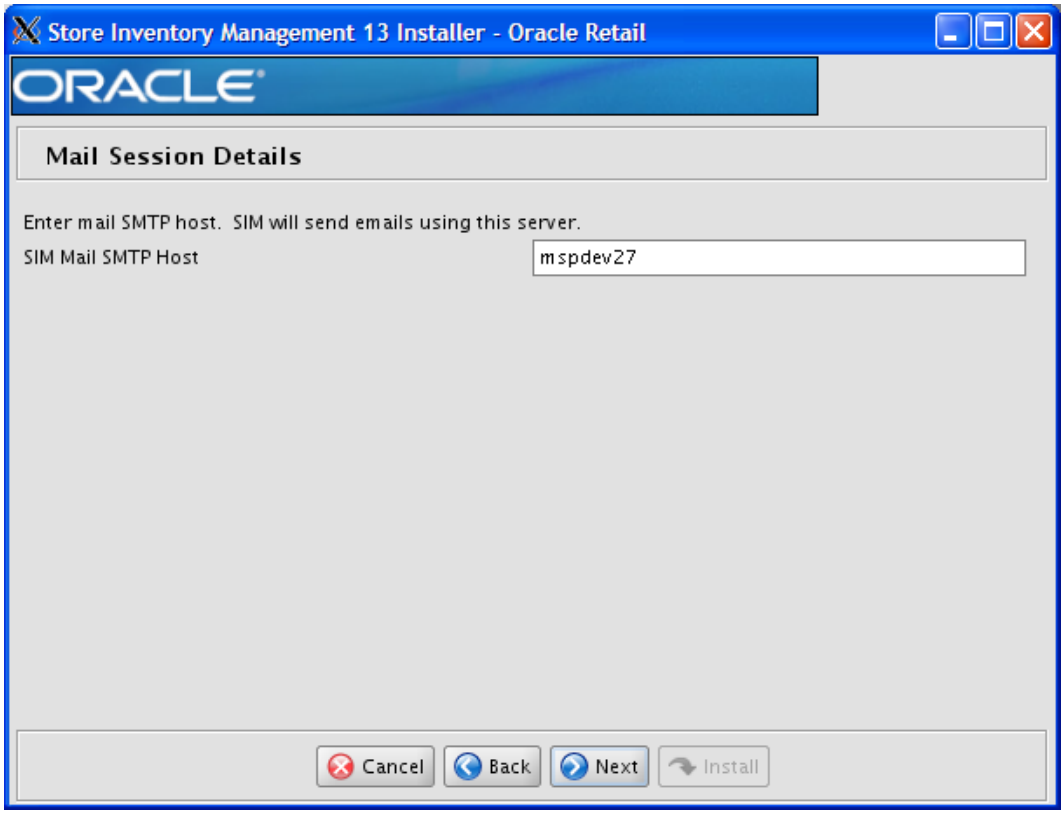
Field Title	LDAP Search Base DN
Field Description	Distinguished name of the LDAP directory entry under which SIM should search for users.
Destination	ldap.cfg
Example	cn=Users,dc=mycompany,dc=com

Field Title	Search User DN
Field Description	Distinguished name of the user that SIM will use to authenticate to the LDAP directory.
Destination	ldap.cfg
Example	cn=admin,dc=mycompany,dc=com

Field Title	Search User Password
Field Description	Password for the search user DN.
Destination	ldap.cfg

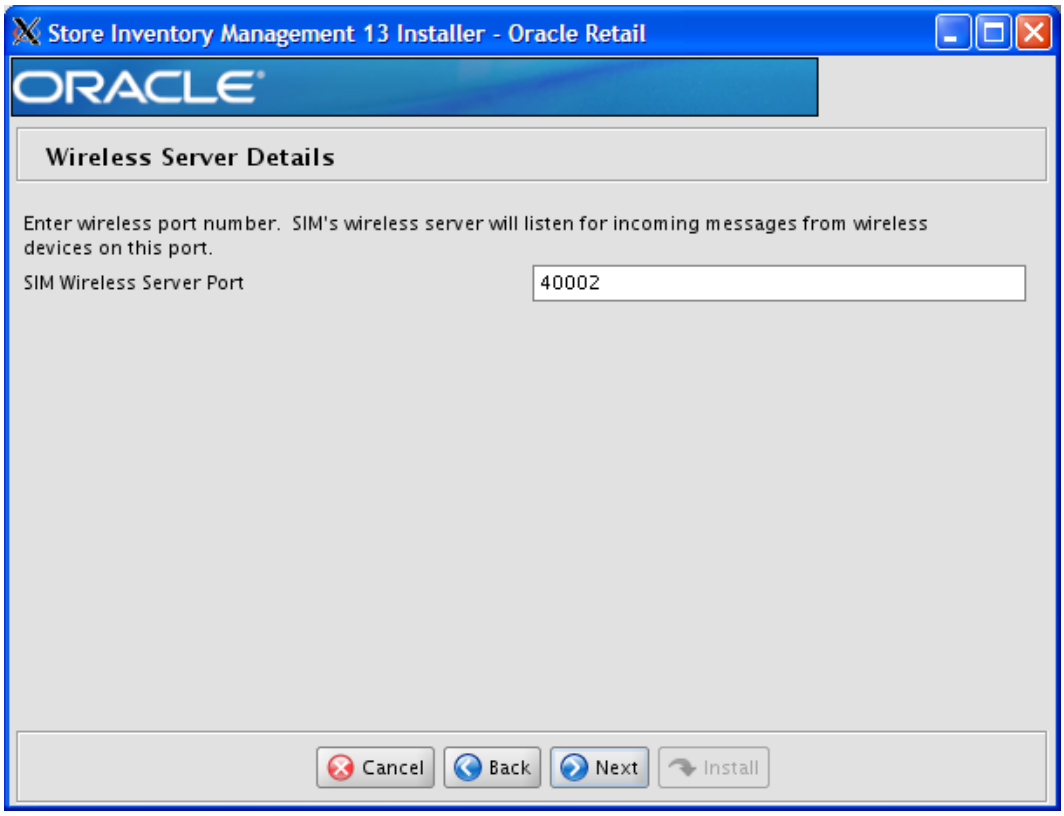
Note: For additional information about LDAP configuration see the *Oracle Retail Store Inventory Management Implementation Guide*.

Screen: Mail Session Details



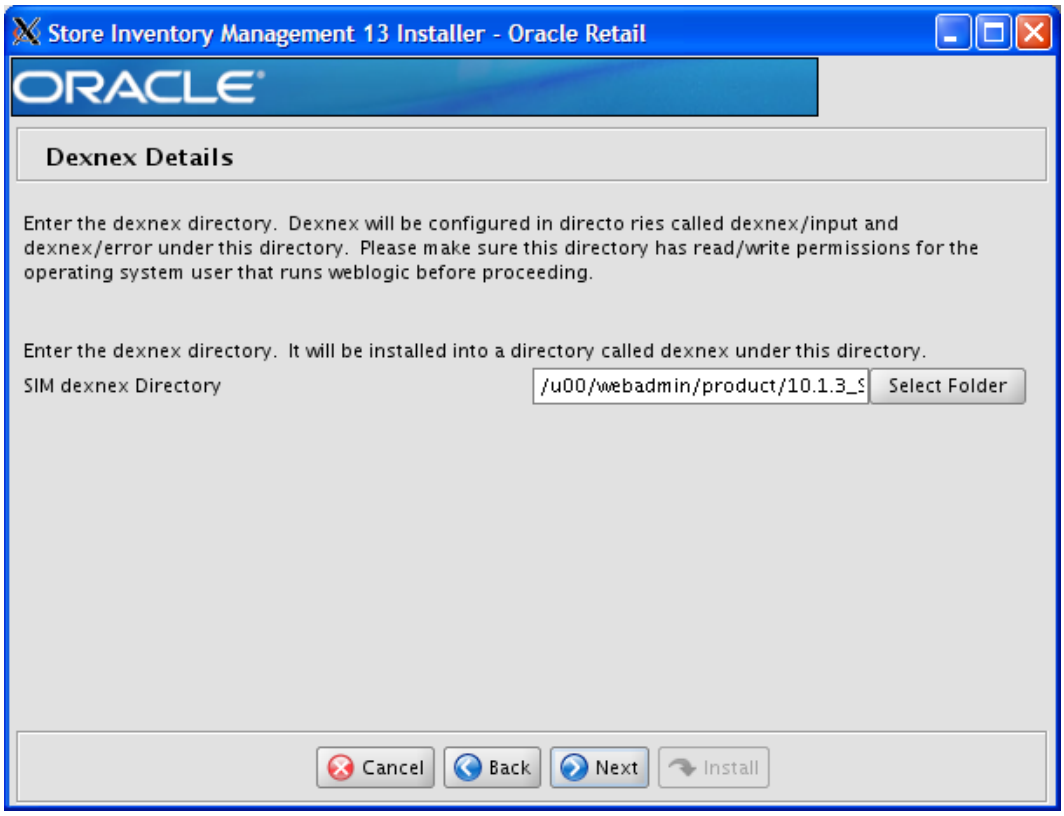
Field Title	SIM Mail SMTP Host
Field Description	Enter mail SMTP host. SIM will send emails using this server.
Destination	mail.cfg
Example	smtp.yourmailhost.com

Screen: Wireless Server Details



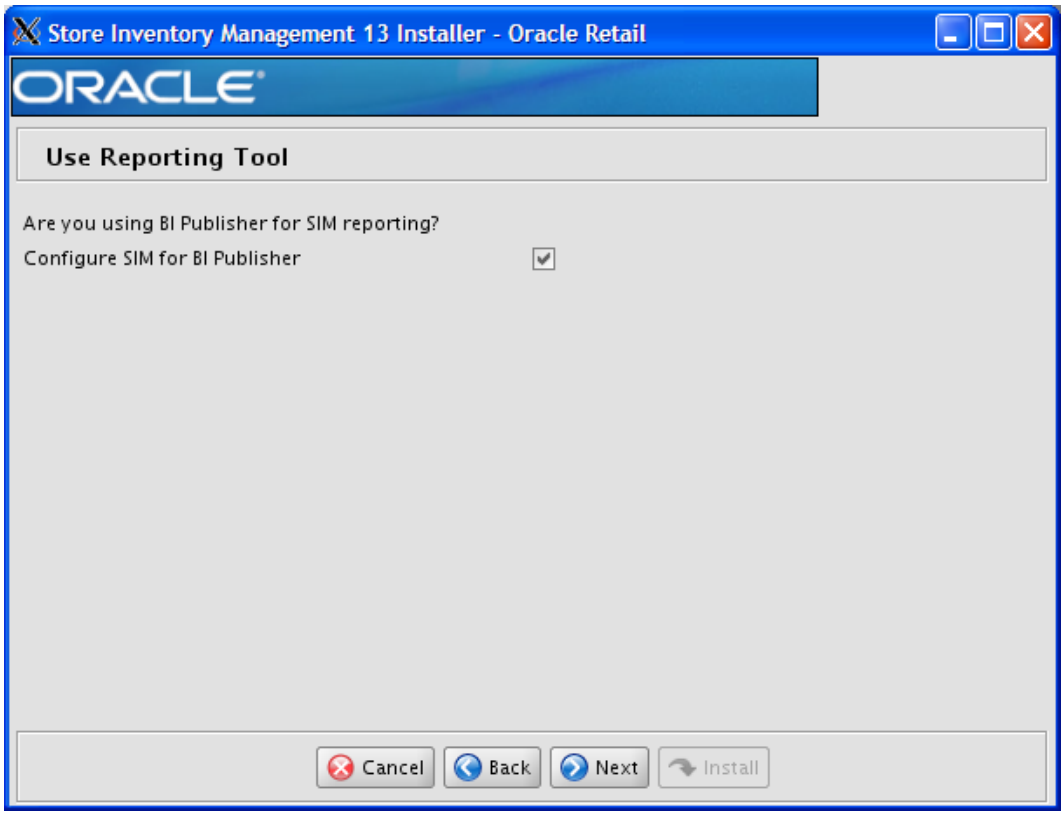
Field Title	SIM Wireless Server Port
Field Description	Choose an available port that the Wavelink server will use to listen for incoming messages from wireless devices.
Destination	wireless_services.cfg, wavelink-startup.sh
Example	40002

Screen: Dexnex Details



Field Title	SIM dexnex Directory
Field Description	The dexnex directory
Example	/u00/webadmin/product/10.1.3_9/OAS/user_projects/domains/java_domain

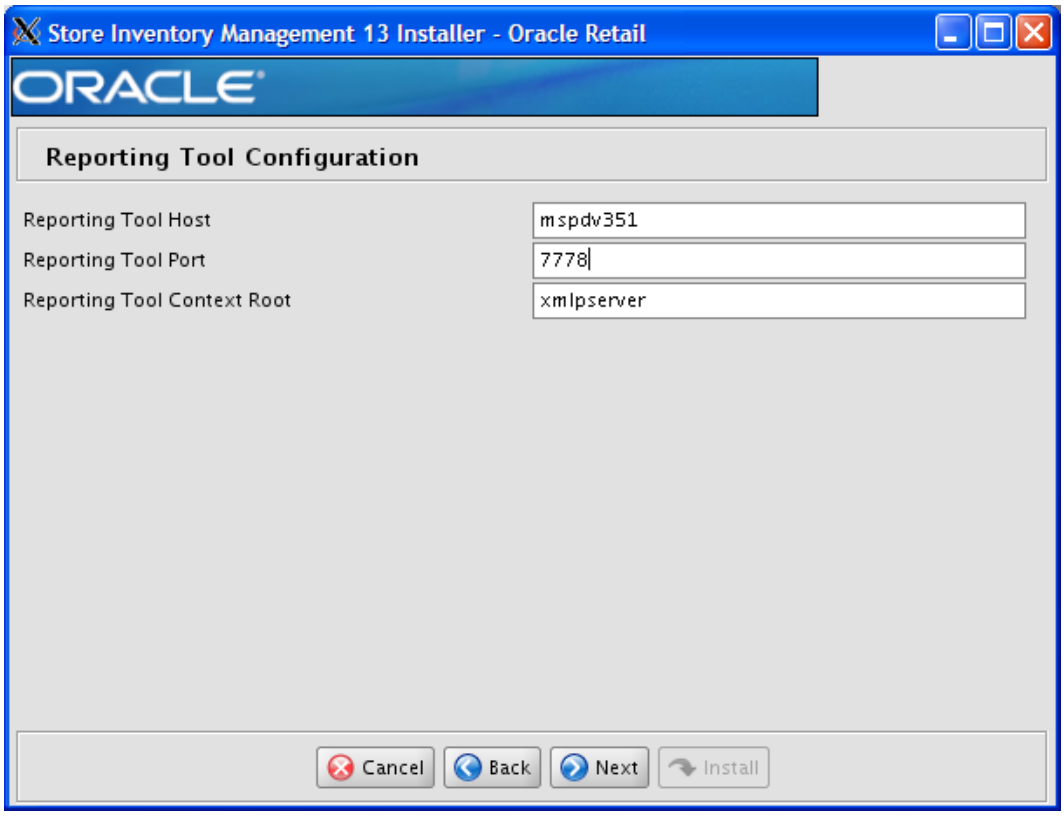
Screen: Use Reporting Tool



Note: See the *Oracle Retail Store Inventory Management Implementation Guide* for SIM reports installation details. If SIM reports will be installed at a later time, leave the reporting configuration values as the default values. They can be configured using the Store and Reporting Tool at a later time.

Field Title	Configure SIM for BI Publisher
Field Description	Toggle field indicating whether or not to configure SIM for BI Publisher Reporting Tool
Destination	insert_default_st_config_val.pls
Example	true
Notes	The following configuration screens will only appear if this checkbox is marked.

Screen: Reporting Tool Configuration

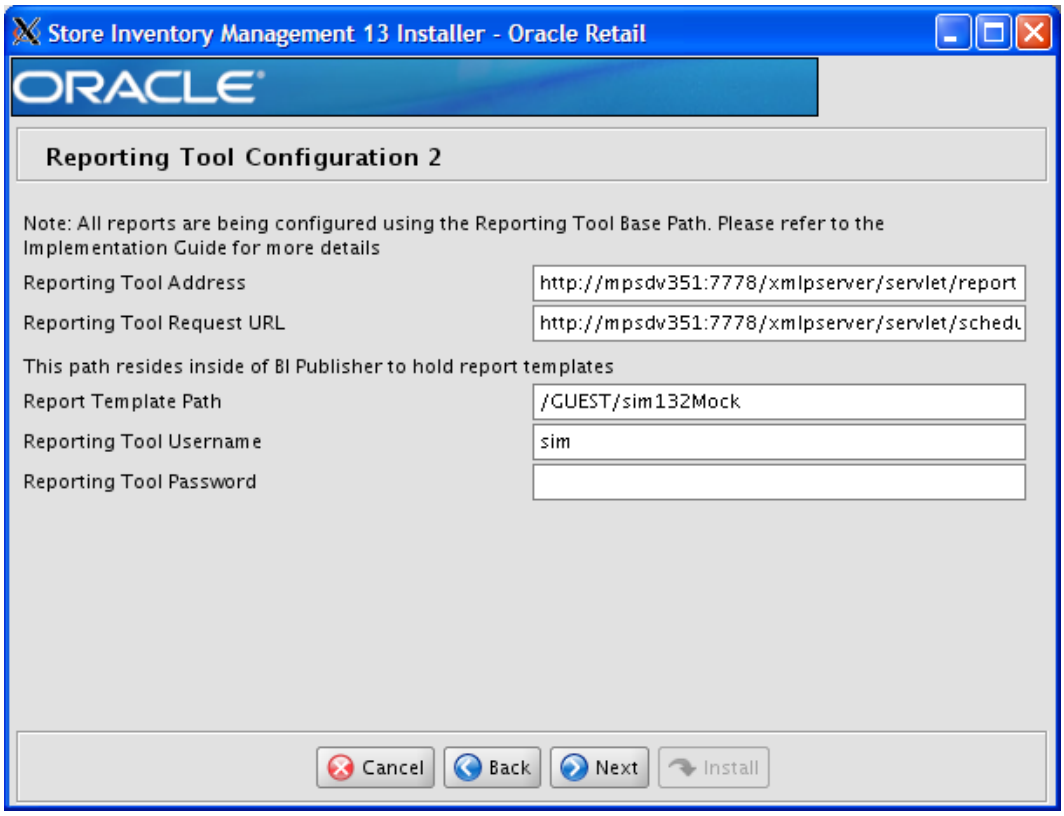


Field Title	Reporting Tool Host
Field Description	Host name where Reporting Tool is installed.
Destination	Updates the reporting tool related default values in SIM database.
Example	myhost.us.oracle.com

Field Title	Reporting Tool Port
Field Description	Port where Reporting Tool is configured.
Destination	Updates the reporting tool related default values in SIM database.
Example	7777

Field Title	Reporting Tool Context Root
Field Description	Context root where Reporting Tool is installed
Destination	Updates the reporting tool related default values in SIM database.
Example	xmlpserver

Screen: Reporting Tool Configuration 2



Field Title	Reporting Tool Address
Field Description	Confirmation field of address configured from values provided on previous screen
Destination	Updates the reporting tool related default values in SIM database.
Example	http://myhost.us.oracle.com:7777/xmlpserver/servlet/report

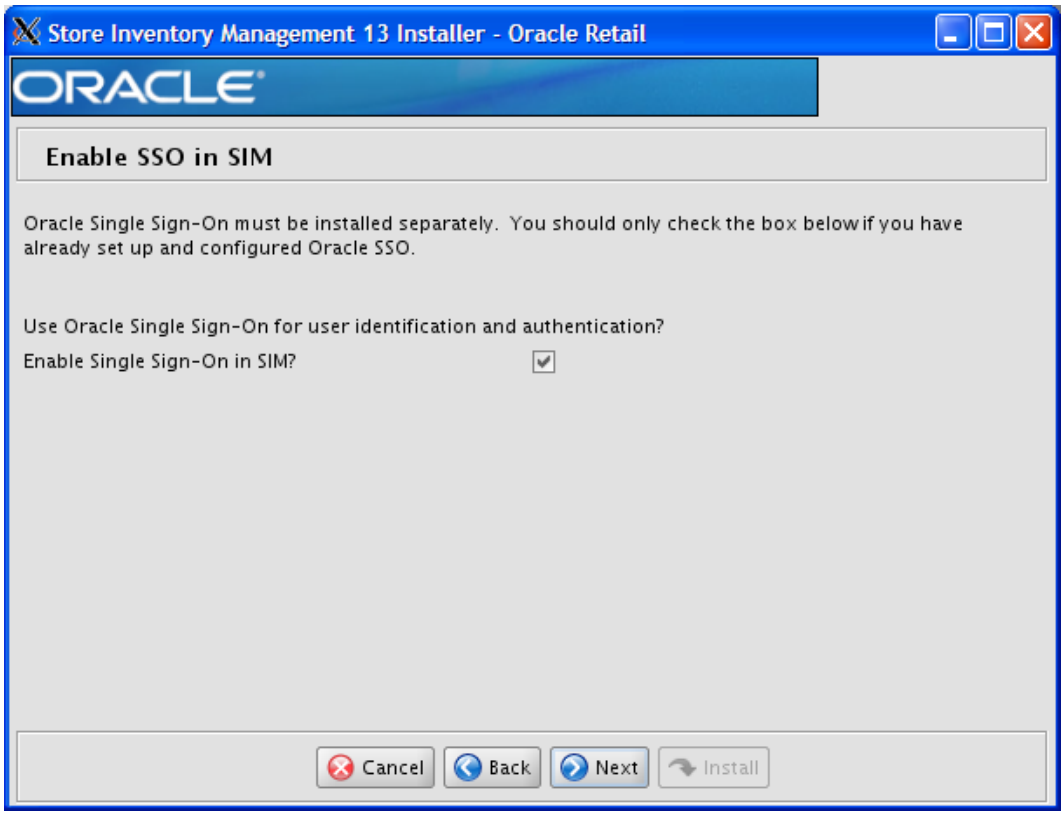
Field Title	Reporting Tool Request URL
Field Description	Confirmation field of address configured from values provided on previous screen
Destination	Updates the reporting tool related default values in SIM database.
Example	http://myhost.us.oracle.com:7777/xmlpserver/servlet/scheduler

Field Title	Reporting Template Path
Field Description	The folder where SIM report templates have been uploaded on the BI Publisher server. For example, if they have been uploaded in the Guest folder, it is /Guest.
Destination	Updates the reporting tool related default values in SIM database.
Example	/Base/SIM/13.2

Field Title	Reporting Tool Username
Field Description	*<BIP_REPORTS_USER>* or *<OSSO_USER>*
Destination	Updates the reporting tool related default values in SIM database.
Example	my username

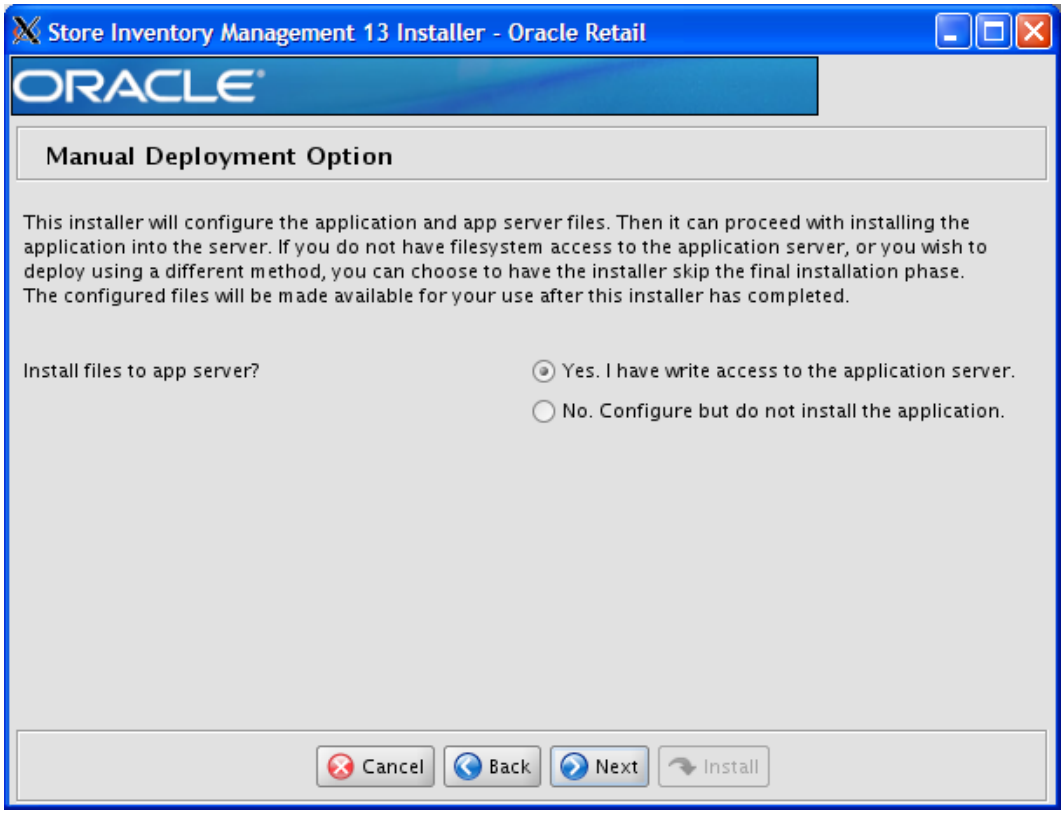
Field Title	Reporting Tool Password
Field Description	*<BIP_REPORTS_USER_PASSWORD>* or *<OSSO_PASSWORD>*
Destination	Updates the reporting tool related default values in SIM database.

Screen: Enable SSO in SIM



Field Title	Enable Single Sign-On in SIM?
Field Description	Configures SIM to enable/disable SSO
Destination	JnlpLaunch.properties

Screen: Manual Deployment Options



Field Title	Install files to app server?
Field Description	If you are running the installer as a user who doesn't have permissions to write to the filesystem under the ORACLE_HOME, then choose "No" to have the installer perform all the configuration within the staging directory but not install any files into the ORACLE_HOME.
Notes	If you choose No, you must perform manual steps to complete the installation.

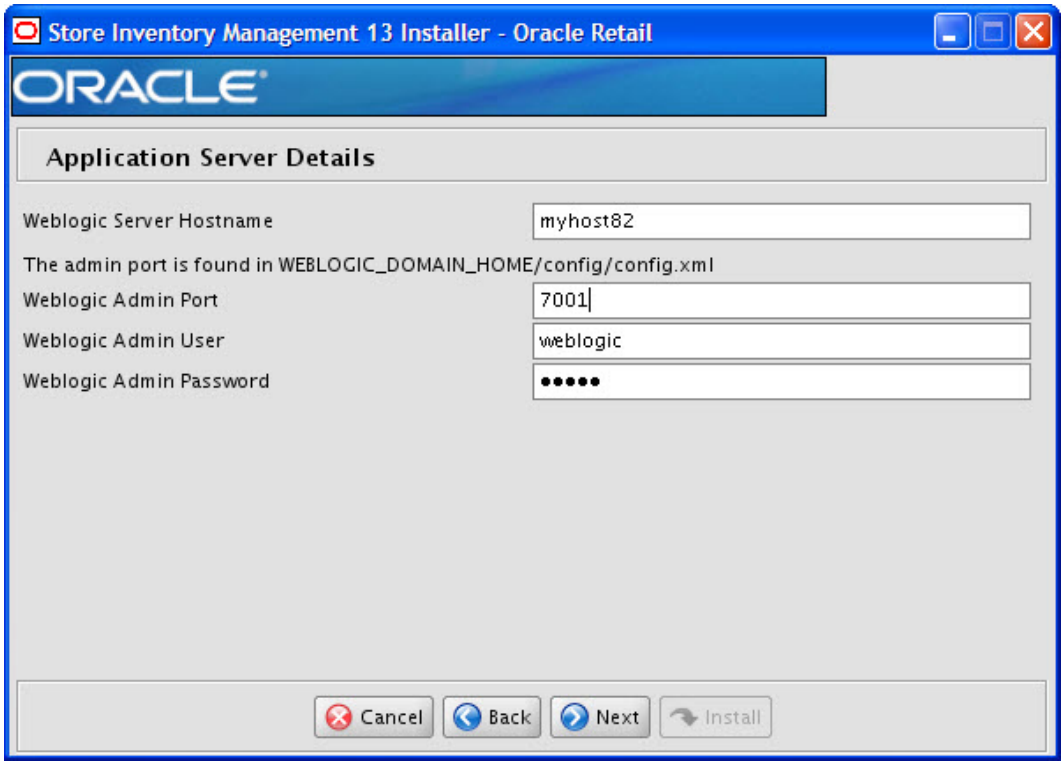
Note: Following the SIM installation, some of the configuration files need to be updated. See the Configuration Files section of the *Oracle Retail Store Inventory Management Operations Guide* for configuration file locations.

C

Appendix: SIM Application WebLogic Server Installer Screens

You need the following details about your environment for the installer to successfully deploy the SIM application. Depending on the options you select, you may not see some screens.

Screen: Application Server Details



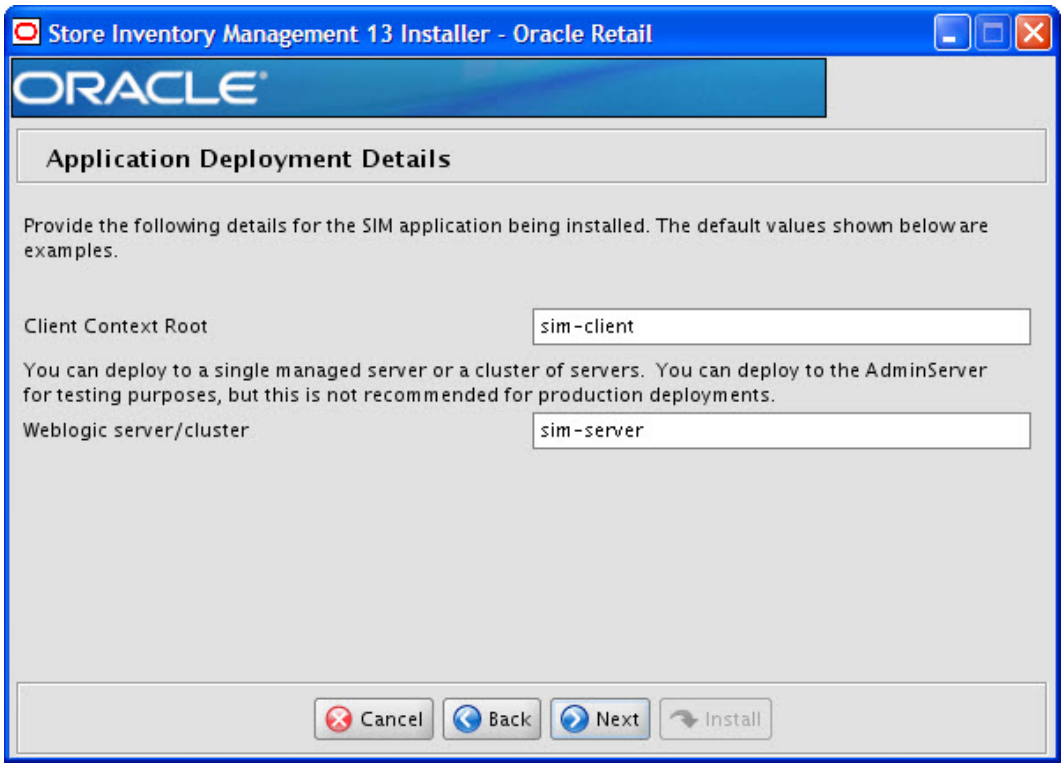
Field Title	WebLogic Server Hostname
Field Description	The hostname of the server where the WebLogic server is installed
Destination	client.cfg, jnlplaunch.properties
Example	Myhost82
Notes	Used by installer scripts to deploy EAR and WAR files and to create default inputs for client codebase and JNDI provider URL

Field Title	WebLogic Admin Port
Field Description	This is the port of Admin console for the domain.
Example	7001
Notes	Used by installer scripts to deploy EAR and WAR files and to create default input for JNDI provider URL

Field Title	WebLogic Admin User
Destination	jndi.cfg, jnlplaunch.properties
Example	weblogic
Notes	Used by installer scripts to deploy EAR and WAR files

Field Title	WebLogic Admin Password
Field Description	The password of the WebLogic Admin User
Destination	jndi.cfg, jnlplaunch.properties
Notes	Used by installer scripts to deploy EAR and WAR files

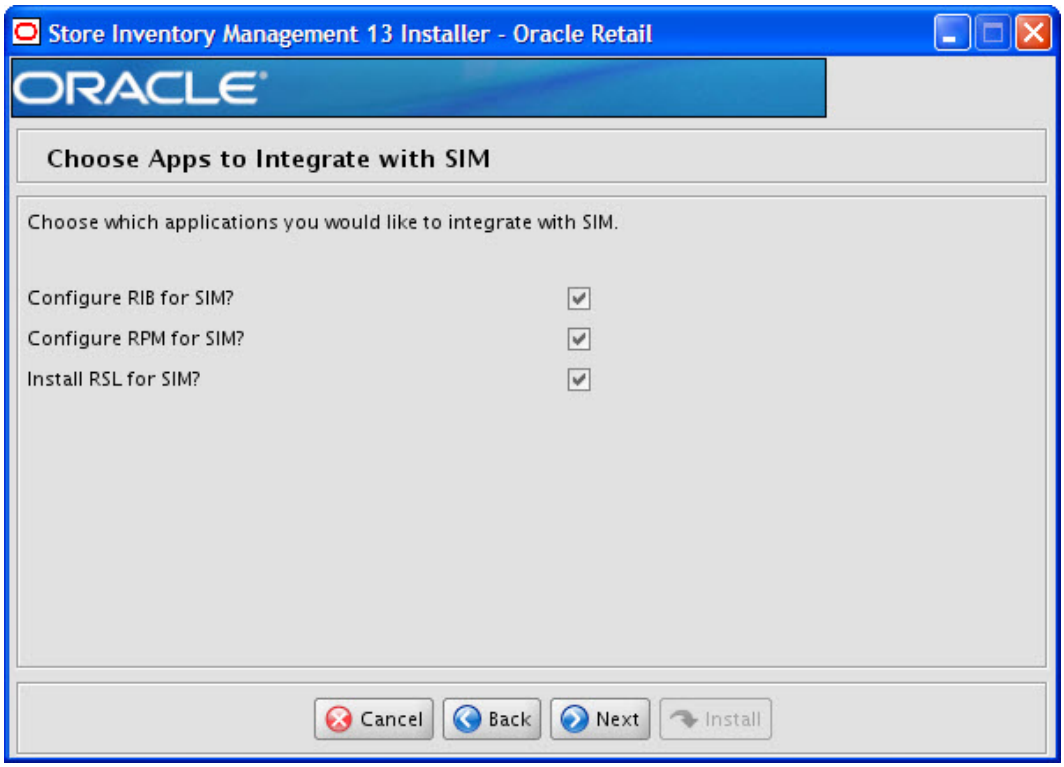
Screen: Application Deployment Details



Field Title	Client Context Root
Field Description	Context root for sim client
Example	sim-client

Field Title	WebLogic server/cluster
Field Description	This the managed server name for standalone deployment and Cluster name for deployment to clustered managed servers
Example	sim-server

Screen: Choose Apps to Integrate with SIM

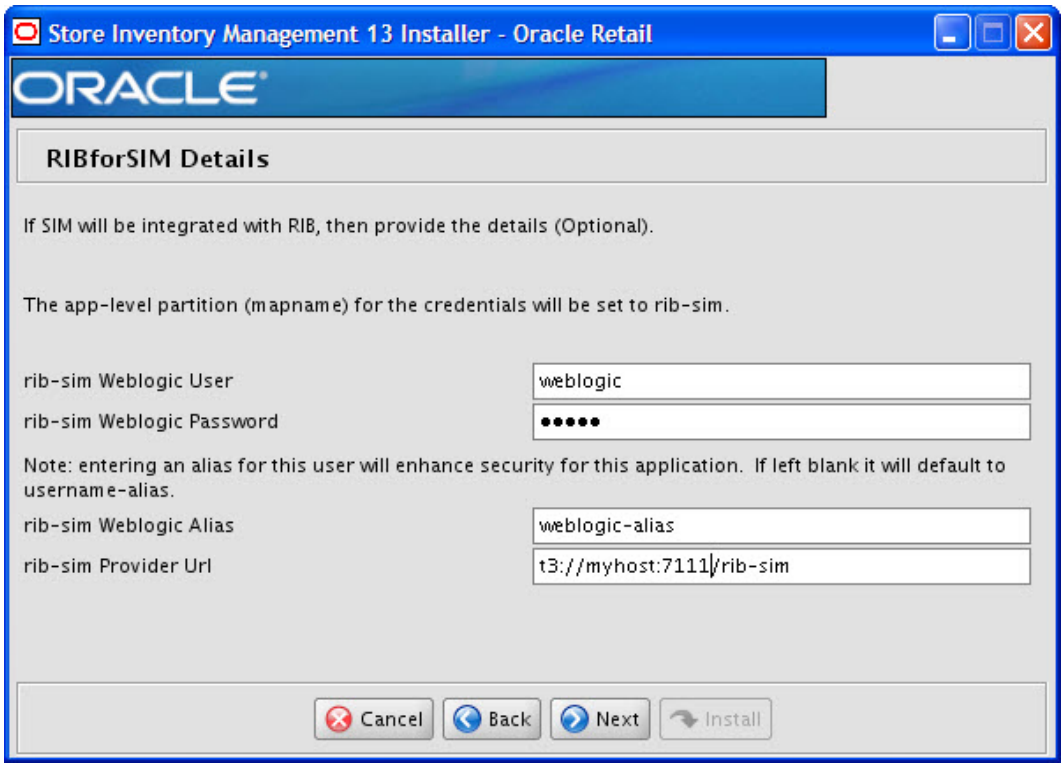


Field Title	Configure RIB for SIM?
Field Description	Select this option if you will be using RIB with SIM.

Field Title	Configure RPM for SIM?
Field Description	Select this option if you will be using RPM with SIM.

Field Title	Install RSL for SIM?
Field Description	Select this option if you will be using RSL with SIM.

Screen: RIBforSIM Details



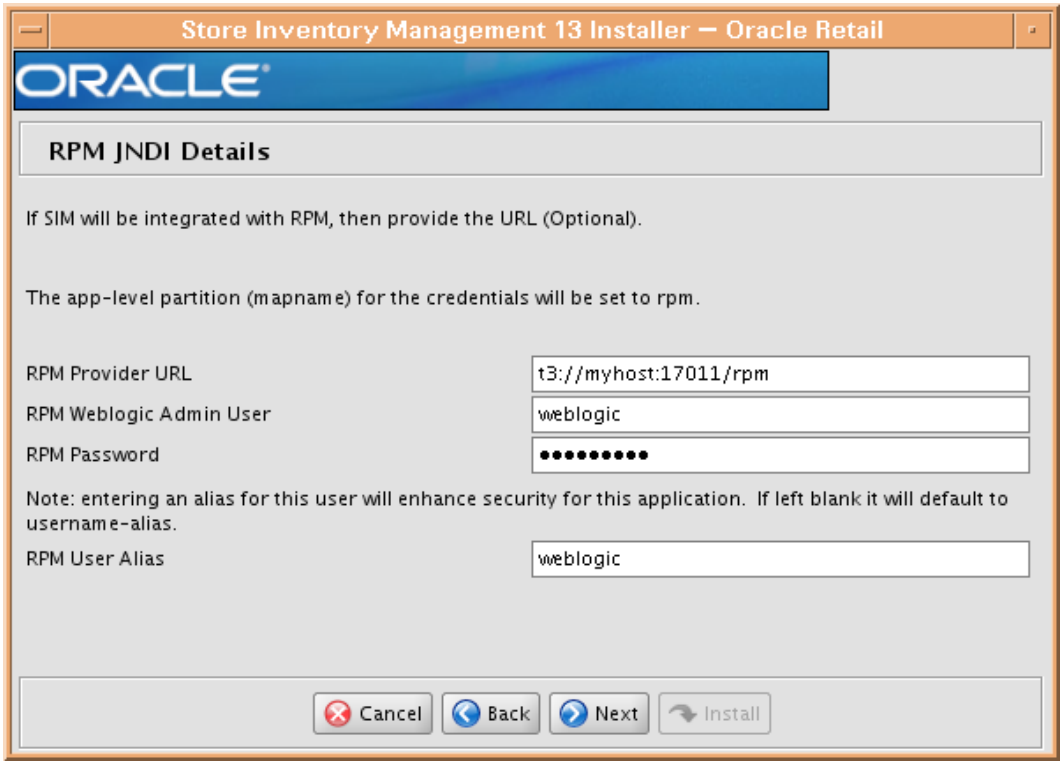
Field Title	rib-sim WebLogic User
Field Description	This is the user name with access to Admin console
Destination	remote_service_locator_info_ribclient.xml
Example	weblogic

Field Title	rib-sim WebLogic Password
Field Description	Password for the RIBforRPM 13 user.

Field Title	rib-sim WebLogic Alias
Field Description	This is the alias for the user name.
Example	weblogic-alias

Field Title	rib-sim Provider URL
Field Description	This the provider URL of the rib-<app>
Examples	t3://myhost:7111/rib-sim

Screen: RPM JNDI Details



Field Title	RPM Provider URL
Field Description	This is the provider URl for <app>
Destination	jndi_providers.xml
Example	t3://myhost:7111/rpm

Field Title	RPM Weblogic Admin User
Field Description	This is the user which has access to RPM Weblogic server.
Example	weblogic

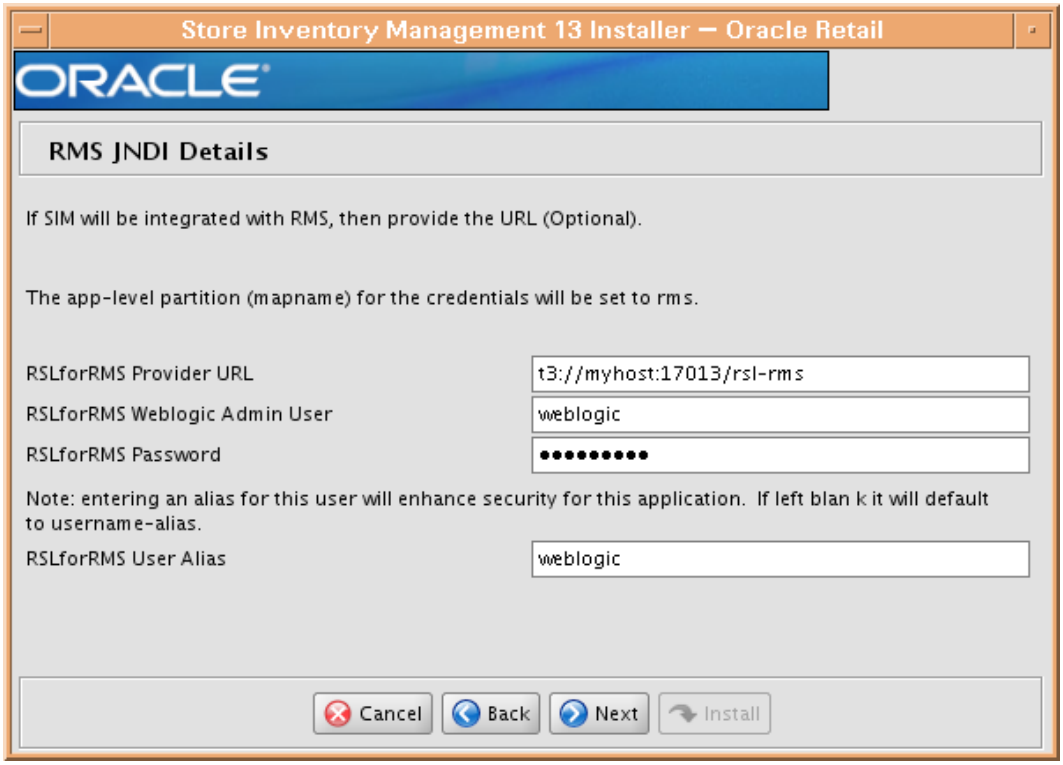
Field Title	RPM Password
Field Description	This is the password of the user provided for RPM Weblogic Admin user in the above.

Field Title	RPM User Alias
Field Description	This is the alias for <RPM Weblogic Admin User>
Examples	weblogic (Make sure to give the same name as provided for <RPM Weblogic Admin User>

Note: In SIM Database, verify that rk_config table has a valid RPM application login user name for config_key=RPM_APP_USER_NAME, RPM_APP_USER_FIRST_NAME and RPM_APP_USER_LAST_NAME.
This is required for SIM-RPM integration to work properly.

For example, retail.user (validate this user by logging into the RPM application).

Screen: RMS JNDI Details



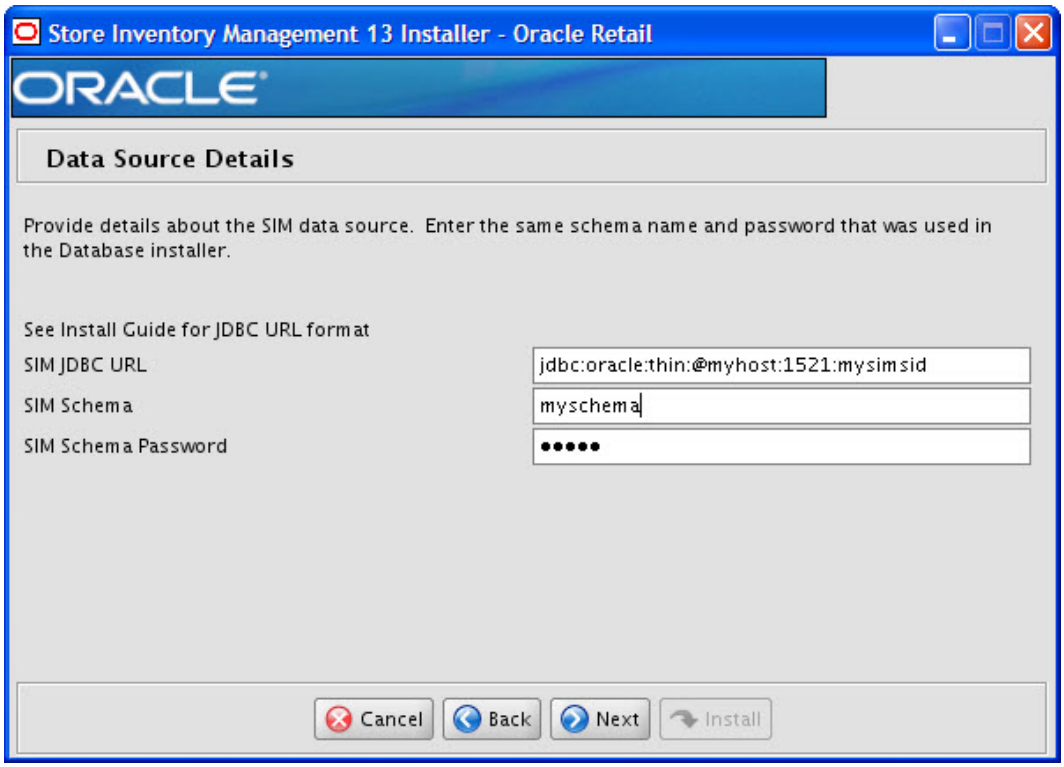
Field Title	RSLforRMS Provider URL
Field Description	This is the provider URL for the RSLforRMS
Destination	jndi_providers.xml
Example	t3://myhost:7111/

Field Title	RSLforRMS Weblogic Admin User
Field Description	This is the user name for login to RSLforRSM Weblogic Server.
Example	weblogic

Field Title	RSLforRMS Password
Field Description	This is the password of the user provided for RSLforRMS Weblogic Admin user in the above.

Field Title	RSLforRMS User Alias
Field Description	This is the alias for RSLforRMS Weblogic Admin User.
Examples	weblogic (Make sure to give the same name as provided for <RSLforRMS Weblogic Admin User>

Screen: Data Source Details

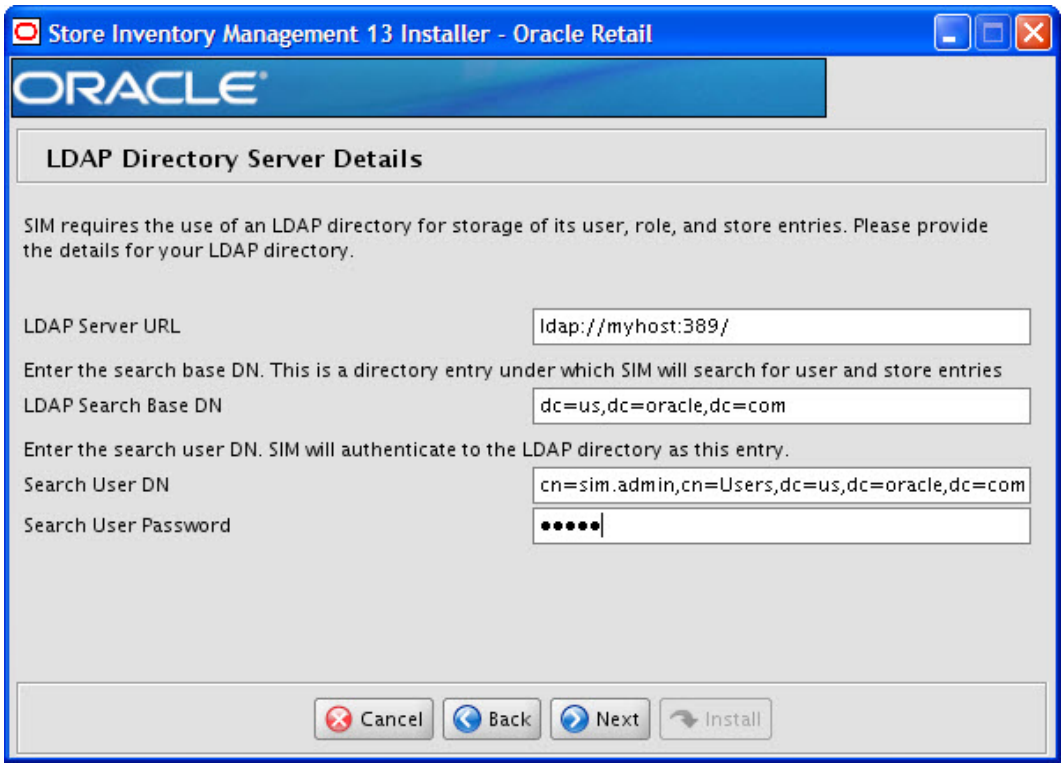


Field Title	SIM JDBC URL
Field Description	URL used by the SIM application to access the SIM database schema.
Destination	WebLogic admin server
Example	jdbc:oracle:thin:@myhost:1521:mysimsid jdbc:oracle:thin:@(DESCRIPTION =(ADDRESS_LIST =(ADDRESS = (PROTOCOL = TCP)(HOST = myhost1)(PORT = 1521))(ADDRESS = (PROTOCOL = TCP)(HOST = myhost2)(PORT = 1521))(LOAD_BALANCE = yes))(CONNECT_DATA =(SERVICE_NAME = mysimsid)))

Field Title	SIM Schema
Field Description	The schema name
Destination	WebLogic admin server
Notes	The schema name should match the name you provided when you ran the database schema installer.

Field Title	SIM Schema Password
Field Description	The password for the SIM Schema
Destination	WebLogic admin server

Screen: LDAP Directory Server Details



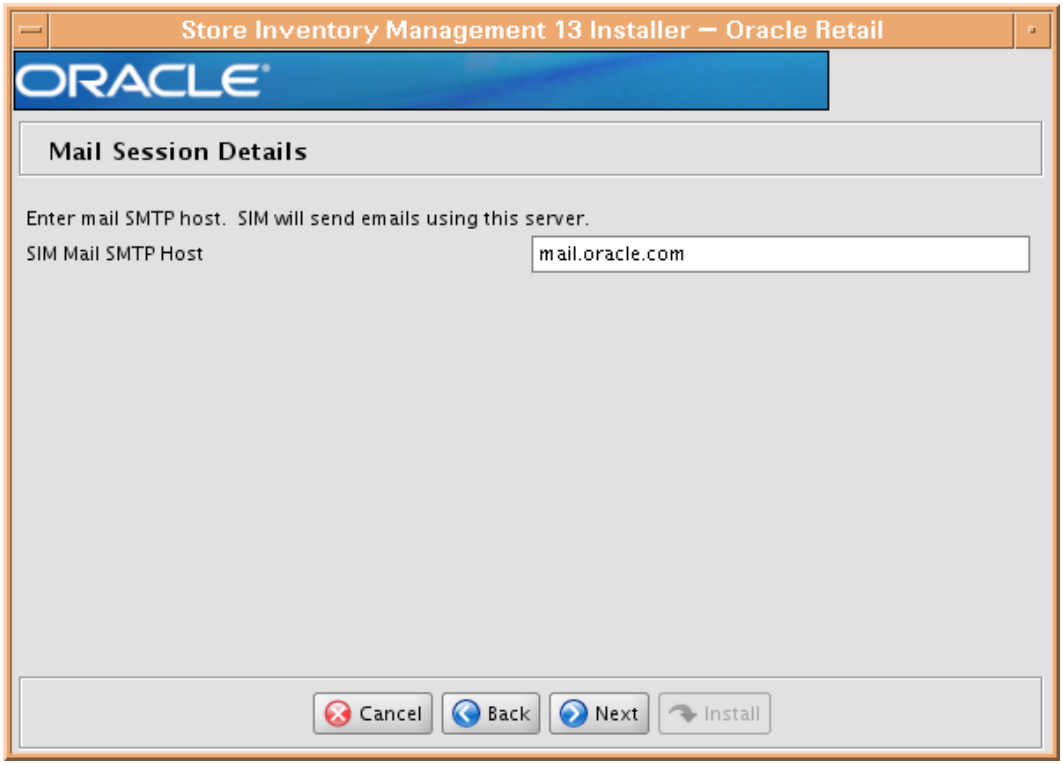
Field Title	LDAP server URL
Field Description	URL for your LDAP directory server.
Destination	ldap.cfg
Example	ldap://myhost:389/

Field Title	LDAP Search Base DN
Field Description	The directory entry under which SIM will search for user and store entries.
Destination	ldap.cfg
Example	dc=us,dc=oracle,dc=com

Field Title	Search User DN
Field Description	Distinguished name of the user that RPM uses to authenticate to the LDAP directory.
Destination	ldap.cfg
Example	cn=admin,dc=mycompany,dc=com

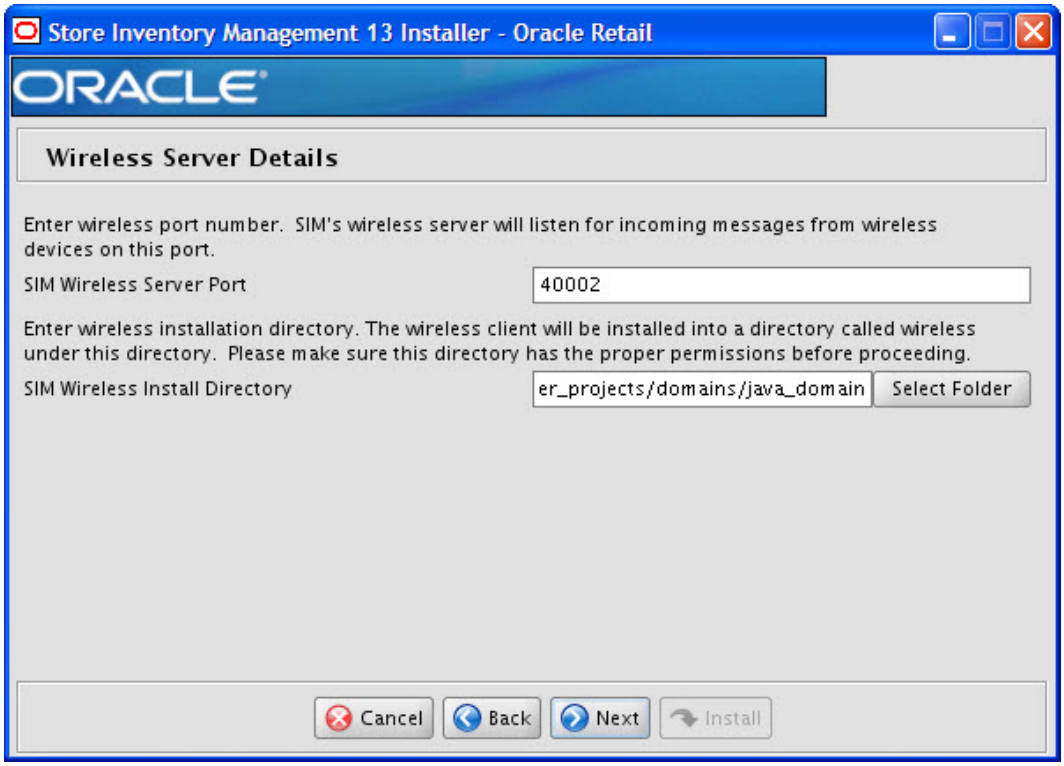
Field Title	Search User Password
Field Description	Password for the search user DN.
Destination	ldap.cfg

Screen: Mail Session Details



Field Title	SIM Mail SMTP Host
Field Description	The SMTP server that will be used to send notification emails from SIM.
Destination	WebLogic admin server
Example	mail.oracle.com

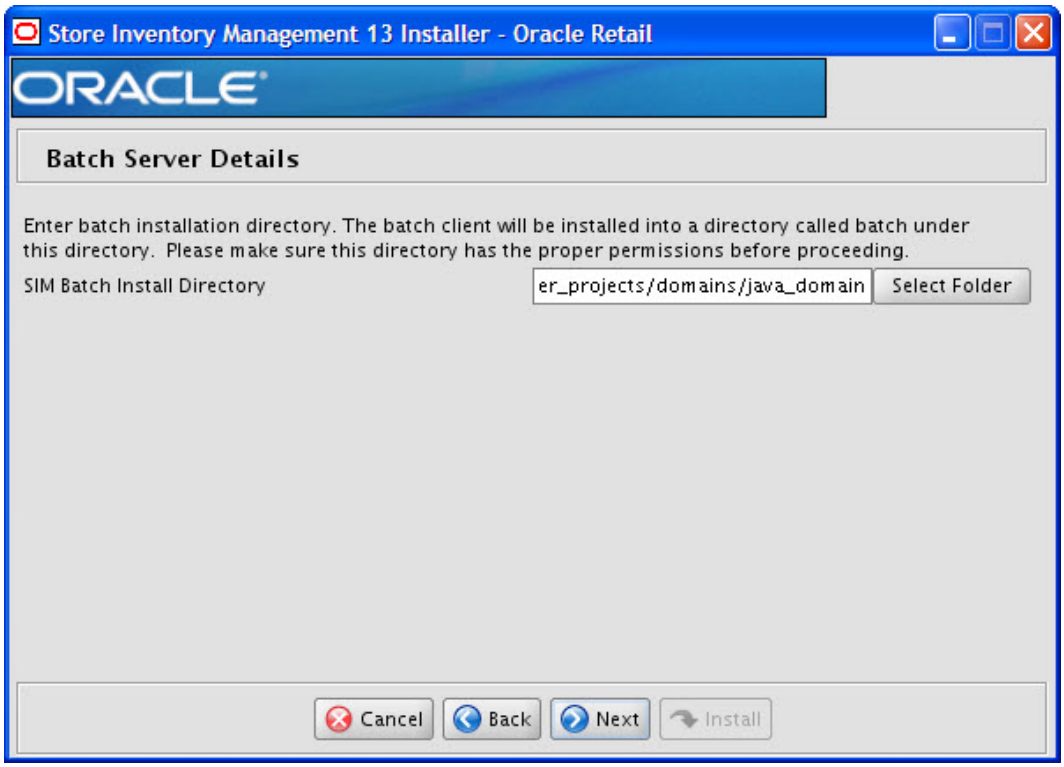
Screen: Wireless Server Details



Field Title	SIM Wireless Server Port
Field Description	Choose an available port that the Wavelink server will use to listen for incoming messages from wireless devices
Destination	wireless.cfg, wavelink-startup.sh
Example	40002

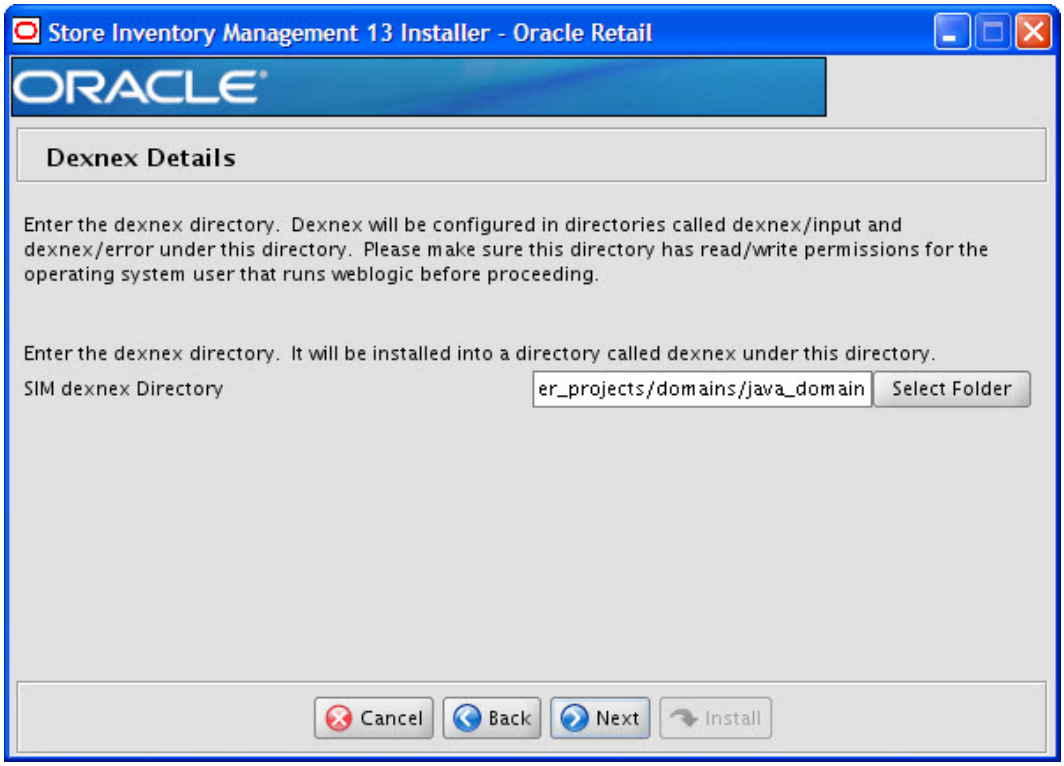
Field Title	SIM Wireless Install Directory
Field Description	The wireless installation directory.
Example	/u00/webadmin/product/10.3.3/WLS/user_projects/domains/java_domain

Screen: Batch Server Details



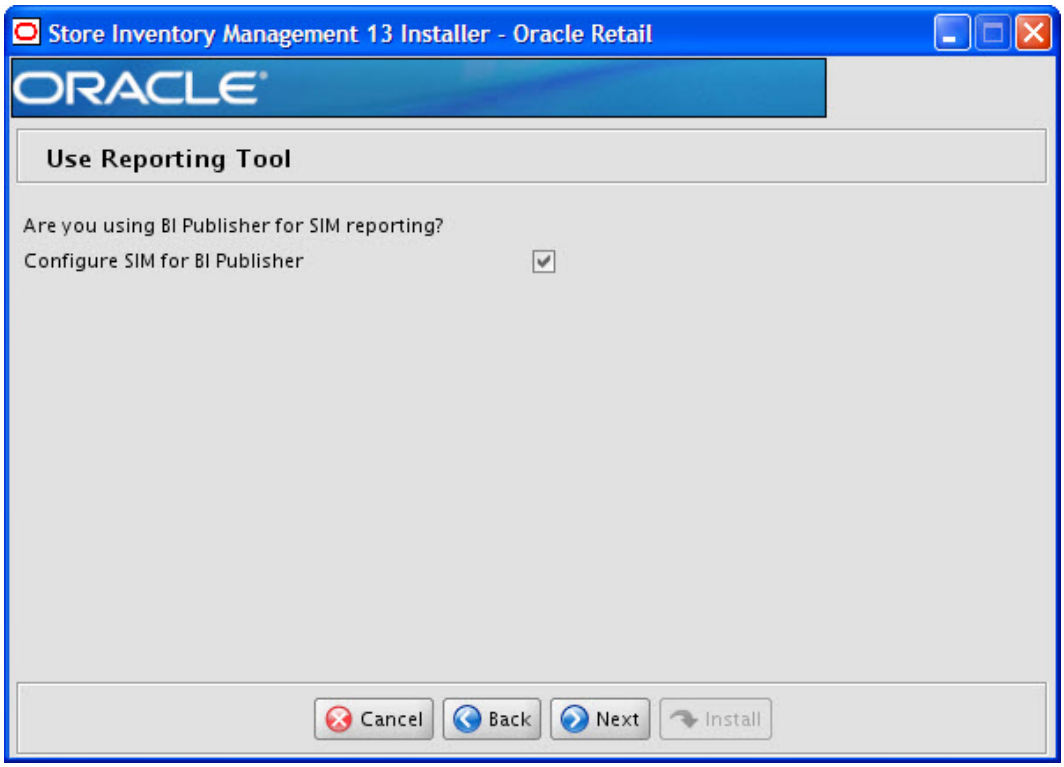
Field Title	SIM Batch Install Directory
Field Description	The batch installation directory.
Example	/u00/webadmin/product/10.3.3/WLS/user_projects/domains/java_domain

Screen: Dexnex Details



Field Title	SIM dexnex Directory
Field Description	The dexnex directory.
Example	/u00/webadmin/product/10.3.3/WLS/user_projects/domains/java_domain

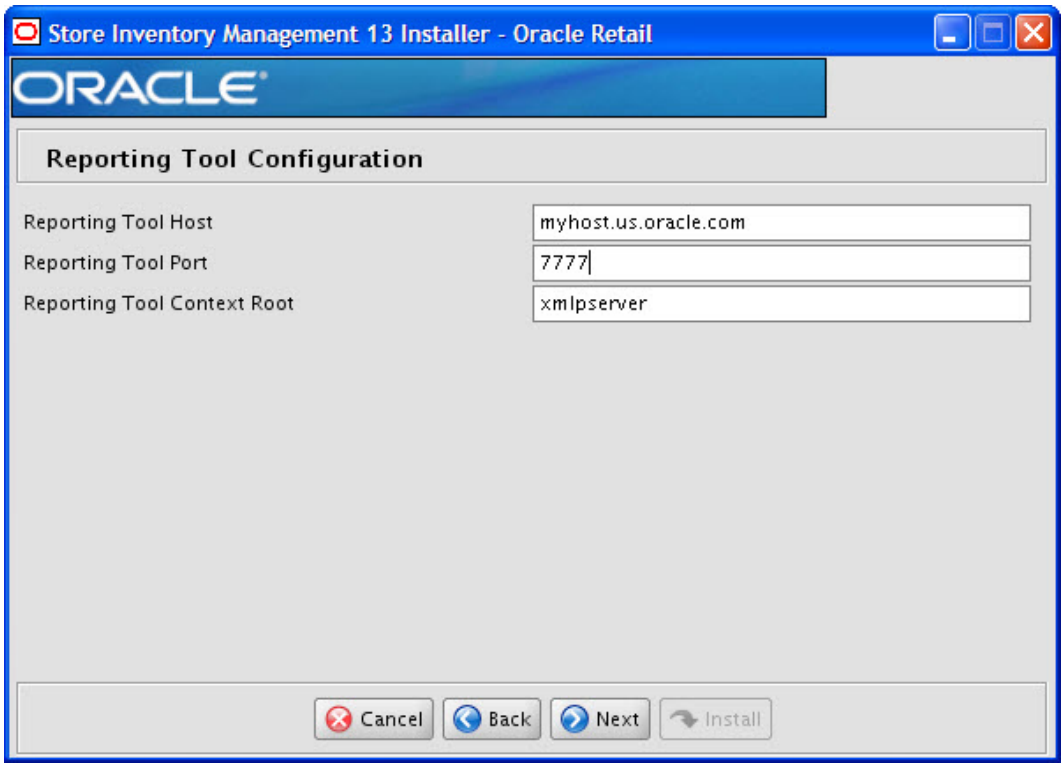
Screen: Use Reporting Tool



Note: See the *Oracle Retail Store Inventory Management Implementation Guide* for SIM reports installation details. If SIM reports will be installed at a later time, leave the reporting configuration values as the default values. These can be configured using the Store and Reporting Tool at a later time.

Field Title	Configure SIM for BI Publisher
Field Description	Toggle field indicating whether or not to configure SIM for BI Publisher Reporting Tool
Destination	insert_default_st_config_val.pls
Example	true
Notes	The following configuration screens will only appear if this checkbox is marked.

Screen: Reporting Tool Configuration

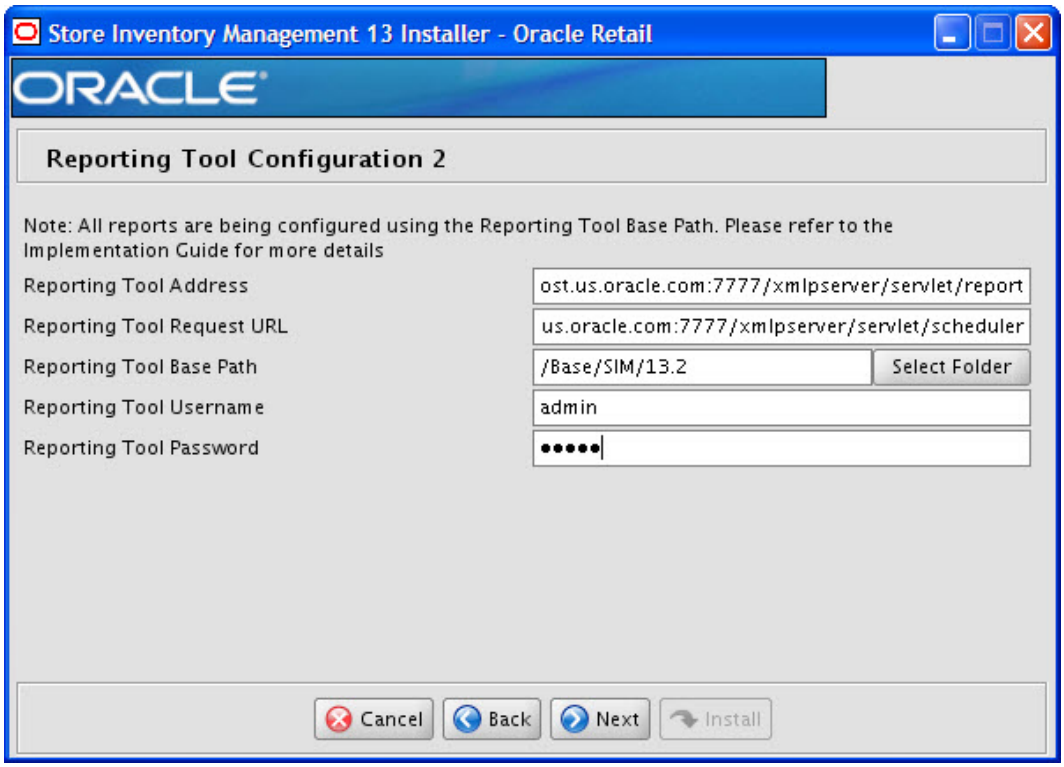


Field Title	Reporting Tool Host
Field Description	Host name where Reporting Tool is installed.
Destination	Updates the reporting tool related default values in SIM database.
Example	myhost.us.oracle.com

Field Title	Reporting Tool Port
Field Description	Port where Reporting Tool is configured.
Destination	Updates the reporting tool related default values in SIM database.
Example	7777

Field Title	Reporting Tool Context Root
Field Description	Context root where Reporting Tool is installed
Destination	Updates the reporting tool related default values in SIM database.
Example	xmlpserver

Screen: Reporting Tool Configuration 2



Field Title	Reporting Tool Address
Field Description	Confirmation field of address configured from values provided on previous screen
Destination	Updates the reporting tool related default values in SIM database.
Example	http://myhost.us.oracle.com:7777/xmlpserver/servlet/report

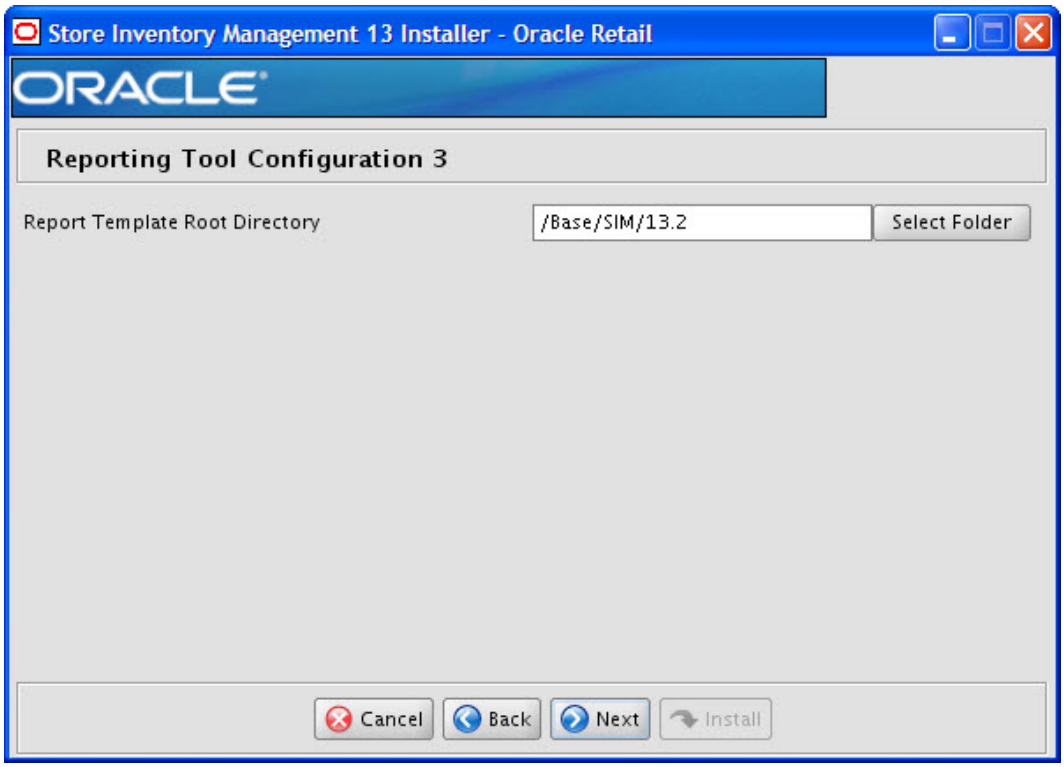
Field Title	Reporting Tool Address URL
Field Description	Confirmation field of address configured from values provided on previous screen
Destination	Updates the reporting tool related default values in SIM database.
Example	http://myhost.us.oracle.com:7777/xmlpserver/servlet/scheduler

Field Title	Reporting Tool Base Path
Field Description	From the *Oracle Retail Store Inventory Management Implementation Guide:* <*BIP_SIM_REPORTS_FOLDER*> is the folder where SIM reports have been uploaded on the BI Publisher server. For example, if they have been uploaded in the Guest folder, it is /Guest.
Destination	Updates the reporting tool related default values in SIM database.
Example	/Base/SIM/13.2

Field Title	Reporting Tool Username
Field Description	From the *Oracle Retail Store Inventory Management Implementation Guide:* <*BIP_REPORTS_USER*> or <*OSSO_USER*>
Destination	Updates the reporting tool related default values in SIM database.
Example	my username

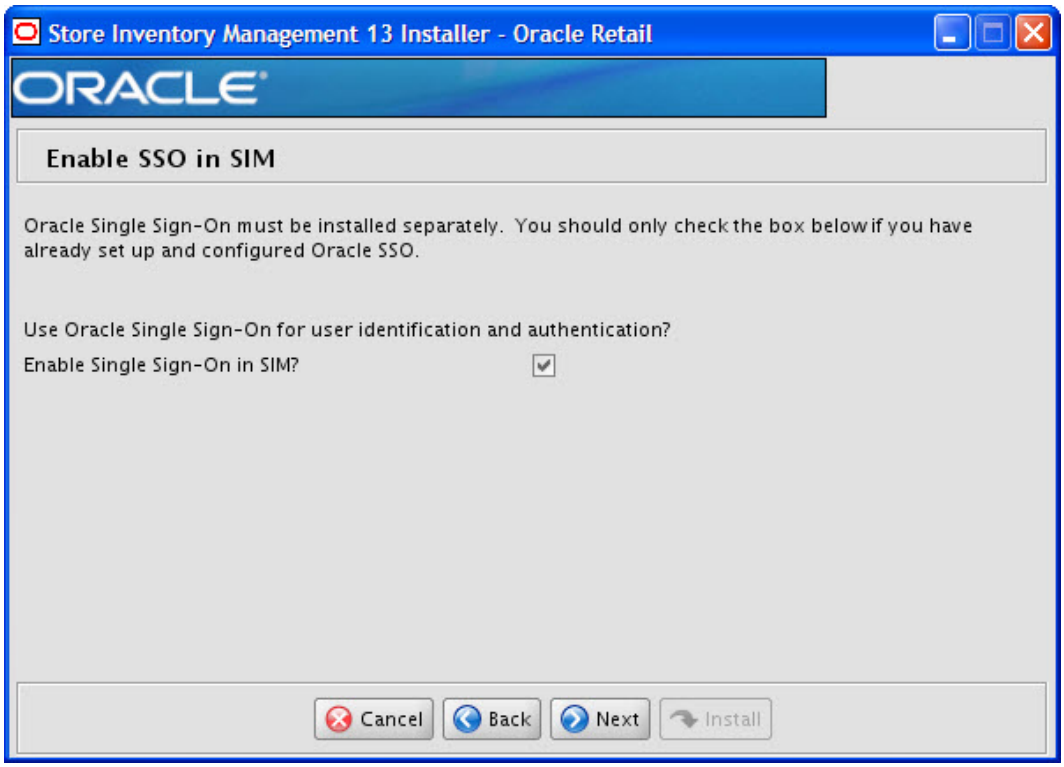
Field Title	Reporting Tool Password
Field Description	From the *Oracle Retail Store Inventory Management Implementation Guide:* <*BIP_REPORTS_USER_PASSWORD*> or <*OSSO_PASSWORD*>
Destination	Updates the reporting tool related default values in SIM database.

Screen: Reporting Tool Configuration 3



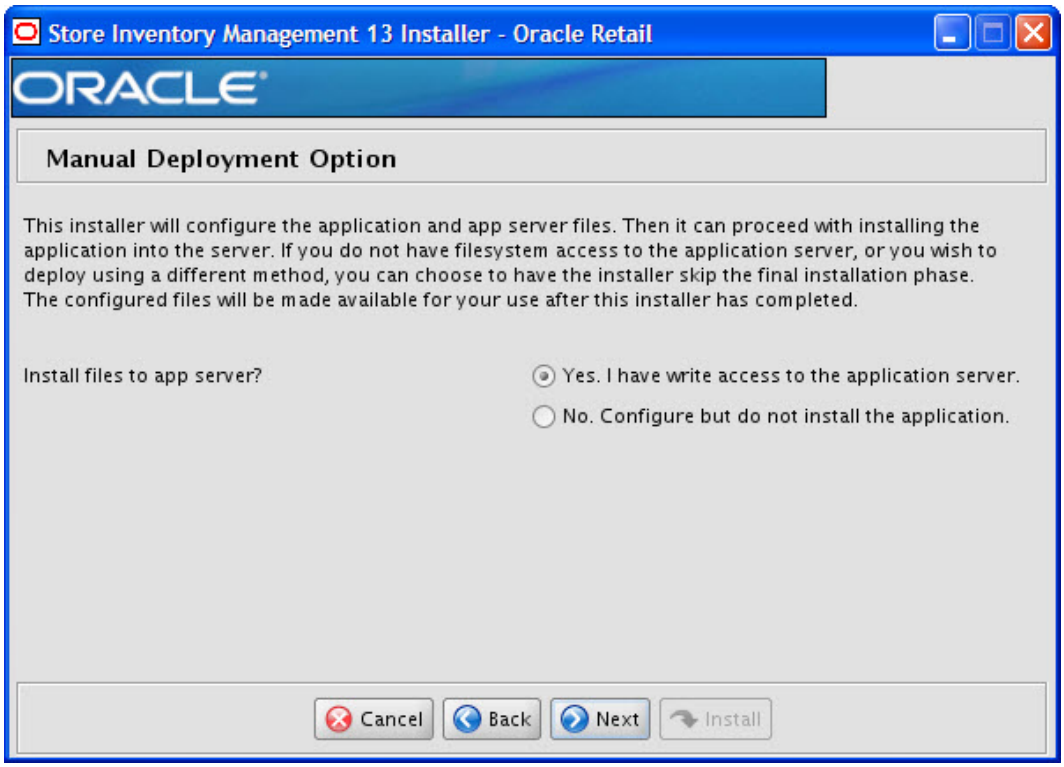
Field Title	Report Template Root Directory
Field Description	The root directory in which your SIM report templates are located.
Example	/Base/SIM/13.2

Screen: Enable SSO in SIM



Field Title	Use Oracle Single Sign-On for user identification and authentication?
Field Description	This version of SIM has the option to use Oracle Single Sign-On (OSSO) technology to authenticate users. If OSSO is being used in your environment, choose Yes. A choice of No will configure SIM to use its own LDAP directory settings for authentication.
Destination	JnlpLaunch.properties

Screen: Manual Deployment Option



Fields on this screen:

Field Title	Install files to app server?
Field Description	If you do not have write access under ORACLE_HOME, you can still use the installer to gather your settings and configure the SIM files locally in the staging area. Then, at a later time, an administrator can manually copy over the SIM files and deploy the ear file. If you select this option, instructions are printed to the console and the installer log file for the steps needed to complete the installation.
Note	Select Yes. There is a known issue when selecting No. If you choose the option, **No. Configure but do not install the application**, in the installer screen named Manual Deployment Option, files required for manual application installation are deleted at the end of the installation

Note: Following the SIM installation, some of the configuration files need to be updated. See the Configuration Files section of the *Oracle Retail Store Inventory Management Operations Guide* for configuration file locations.

D

Appendix: Installer Silent Mode

Repeating an Installation Attempt

In addition to the GUI and text interfaces of the installer, there is a silent mode that can be run. This mode is useful if you wish to run a repeat installation without retyping the settings you provided in the previous installation. It is also useful if you encounter errors in the middle of an installation and wish to continue.

The installer runs in two distinct phases. The first phase involves gathering settings from the user. At the end of the first phase, a properties file named ant.install.properties is created with the settings that were provided. Then the second phase begins, where this properties file is used to provide your settings for the installation.

To skip the first phase and reuse the ant.install.properties file from a previous run, follow these instructions:

1. Edit the ant.install.properties file and correct any invalid settings that may have caused the installer to fail in its previous run.
2. Run the installer again with the silent argument.

Example: install.sh silent

E

Appendix: Common Installation Errors

This section provides some common errors encountered during installation.

EJB Deployment Errors during Installation to WebLogic

Symptom

On servers that are encountering high memory usage, deployment of sim-server.ear will occasionally fail due to WebLogic's inability to start the EJB polling timer service.

```
     [java] .....Failed to deploy the application with status failed
     [java] Current Status of your Deployment:
     [java] Deployment command type: deploy
     [java] Deployment State       : failed
     [java] Deployment Message     : weblogic.application.ModuleException:
Exception activating module: EJBModule(
sim-ejb3.jar)
     [java]
     [java]
     [java] weblogic.management.scripting.ScriptException: Error occured while
performing deploy : Deployment Fail
ed.
     [java] Unable to deploy EJB: PollingCoordinatorThreadBean from sim-ejb3.jar:
     [java]
     [java] Error starting Timer service
```

Solution

Delete the WebLogic managed server/cluster where sim was targeted in the Admin Console, and activate the changes. Manually delete the managed server directory <DOMAIN HOME>/servers/<SIM SERVER NAME>. Bounce the WebLogic admin server. Re-create the managed server in the Admin Console, Finally, re-run the installer. If the error persists after re-installation, consider reducing the cpu, disk, and memory load on the server.

XML Processing Errors While configuring sim-client.ear or sim-server.ear

Symptom

The installer fails while attempting to configure sim-client.ear or sim-server.ear. When updating META-INF/application.xml, the following error occurs:

```
   [mkdir] Created dir:
/work/sources/13.3/SIM_13.2.1/CDROM/WLS/sim/application/sim13/configured-
output/tmp/client/earcontents/afterconfig/META-INF
  [xmltask] It looks like you've got a network error. The probable cause
  [xmltask] is that you're trying to resolve a DTD on the internet although
  [xmltask] you don't know it! Check your XML for DTDs external to your network
  [xmltask] and read the Ant documentation for <xmlcatalog>. XMLTask will support
  [xmltask] usage of <xmlcatalog>. See the following:
  [xmltask] http://ant.apache.org/manual/CoreTypes/xmlcatalog.html
  [xmltask] http://www.oopsconsultancy.com/software/xmltask
  [xmltask] If this isn't the problem, then please report this error to the
support
  [xmltask] mailing list. Thanks!
```

Solution

This error occurs because the server on which SIM is being installed is not able to connect to the internet (for example, java.sun.com). Do either of the following.

- Establish a connection to the internet and re-run the installer, or:
- Perform the following configuration steps:
  i. cd sim/application/sim13/sim-client
  ii. jar xf sim-client.ear META-INF/application.xml
  iii. edit META-INF/application.xml, and remove the <DTD…> tag.
  iv. jar uf sim-client.ear META-INF/application.xml
  v. perform the same steps for sim/application/sim13/sim-server/sim-server.ear.
  vi. re-run the installer.

Output Freezes during Text Mode Installation to OAS and WebLogic

Symptom

The standard output of the installer in text mode will sometimes freeze partway through the installation.

Solution

Open a new terminal to the server and tail the log file located in sim/application/logs.

Database Installer Hangs on Startup

Symptom

When the database schema installer is run, the following is written to the console and the installer hangs indefinitely:

```
Running pre-install checks
Running tnsping to get listener port
```

Solution

The installer startup script is waiting for control to return from the **tnsping** command, but tnsping is hanging. Type Control+C to cancel the installer, and investigate and solve the problem that is causing the **tnsping <sid>** command to hang. This can be caused by duplicate database listeners running.

Unreadable Buttons in the Installer

If you are unable to read the text within the installer buttons, it probably means that your JAVA_HOME is pointed to a pre-1.4.2 JDK. Set JAVA_HOME to a Java development kit of version 1.4.2 or later and run the installer again.

Message: Unable to get a deployment manager

Symptom

The application installer quits with the following error message:

```
[oracle:deploy] Unable to get a deployment manager.
[oracle:deploy]
[oracle:deploy] This is typically the result of an invalid deployer URI format
being supplied, the target server not being in a started state or incorrect
authentication details being supplied.
[oracle:deploy]
[oracle:deploy] More information is available by enabling logging -- please see
the Oracle Containers for J2EE Configuration and Administration Guide for details.
```

Solution

This error can be caused by any of the following conditions:

- OC4J instance provided is not running.
- Incorrect OC4J instance name provided
- Incorrect OC4J administrative username and/or password
- Incorrect OPMN request port provided.

Make sure that the OC4J instance is running, and then check the **ant.install.properties** file for entry mistakes. Pay close attention to the input.deployer.uri, input.oc4j.instance, input.admin.user, and input.admin.password properties. If you need to make a correction, you can run the installer again with this file as input by running silent mode (see Appendix: Installer Silent Mode in this document).

Warning: Could not create system preferences directory

Symptom

The following text appears in the installer Errors tab:

```
May 22, 2006 11:16:39 AM java.util.prefs.FileSystemPreferences$3 run
WARNING: Could not create system preferences directory. System preferences are
unusable.
May 22, 2006 11:17:09 AM java.util.prefs.FileSystemPreferences
checkLockFile0ErrorCode
WARNING: Could not lock System prefs. Unix error code -264946424.
```

Solution

This is related to Java bug 4838770. The /etc/.java/.systemPrefs directory may not have been created on your system. See http://bugs.sun.com for details.

This is an issue with your installation of Java and does not affect the Oracle Retail product installation.

Keystore Errors When Signing sim-config.jar

Symptom

keytool error: java.io.IOException: Keystore was tampered with, or password was incorrect

Solution

This message may be encountered when you use the keytool utility to create an alias for signing the sim-config.jar file. This usually happens when the alias for which you are generating a key already exists in the keystore file.

Delete or rename the ~/.keystore file and run the keytool command again. This creates a fresh keystore file.

Warning: Couldn't find X Input Context

Symptom

The following text appears in the console window during execution of the installer in GUI mode:

```
Couldn't find X Input Context
```

Solution

This message is harmless and can be ignored.

ConcurrentModificationException in Installer GUI

Symptom

In GUI mode, the errors tab shows the following error:

```
java.util.ConcurrentModificationException
            at
java.util.AbstractList$Itr.checkForComodification(AbstractList.java:448)
            at java.util.AbstractList$Itr.next(AbstractList.java:419)
… etc
```

Solution

You can ignore this error. It is related to third-party Java Swing code for rendering of the installer GUI and does not affect the retail product installation.

Error while unpacking the ear file

Symptom

```
The following text appears in the console window during execution of the
installer:

07/12/19 10:53:17 Notification ==>Error while unpacking sim13.ear
java.util.zip.ZipException: error in opening zip file
```

Solution

This is a known bug (BugID 6330834) related to Solaris and NFS in Oracle Application Server 10.1.3.4. Follow the workaround documented for this bug: in the opmn.xml file in $ORACLE_HOME/opmn/conf add the following parameter to the java-options for the instance you are installing.

-Doc4j.autoUnpackLockCount=-1

After making this change you should reload OPMN, restart the affected OC4J instance(s), and retry the retail application installation.

A Second Login Screen Appears After Single Sign-On Login

If you are using Oracle Single Sign-On, you should not need to enter a SIM user name and password once SIM is launched. If the SIM login screen pops up, it means something went wrong with the SSO login. This could be caused by any of the following problems:

- There is no SIM user in LDAP for the SSO user name you are using.
- Permissions are not set up correctly for the SSO user in SIM.
- SSO is configured wrong on the server.
- SSO timed out. (This can happen especially the first time you launch SIM. Try launching SIM again.)

Symptom

A second login screen appears after you have already logged in to Single Sign-On.

Solution

See the *Oracle Retail Store Inventory Management Implementation Guide* for more information on setting up SIM users and using LDAP and SSO with SIM.

Error Connecting to Database URL

Symptom

After entering database credentials in the installer screens and hitting next, a message pops up with an error like this:

```
Error connecting to database URL <url> as user <user> details...
```

The message prevents you from moving on to the next screen to continue the installation.

Solution

This error occurs when the installer fails to validate the user credentials you have entered on the screen. Make sure that you have entered the credentials properly. If you receive a message similar to this:

```
Error connecting to database URL <url> as user <user> java.lang.Exception:
UnsatisfiedLinkError encountered when using the Oracle driver.
Please check that the library path is set up properly or switch to the JDBC thin
client.
```

It may mean that the installer is using the incorrect library path variables for the platform you are installing on. Open the file <STAGING_DIR>/rms/dbschema/common/preinstall.sh and toggle the variable, use32bit, to True if it is set to False or vice versa. This setting is dependant on the JRE that is being used.

Installer Fails because of missing .jar in $ORACLE_HOME/utils/ccr/lib

Symptom

The jar file expected by the installer (emocmclnt.jar) is overwritten after the OPatch patch 6880880 is applied, and any other patch is applied afterward using that OPatch. If you try running the installer after patching, as outlined in the installation guides for forms based applications, the installer fails. All applications that are installed in the same WebLogic server that hosts any of the forms applications are affected by this issue.This is because of required Oracle patches for Linux 64-bit systems that are applied to the forms server.

Solution

Back up the content of the $ORACLE_HOME/utils/ccr/lib directory prior to applying OPatch patch 6880880, and recopy the content back after you apply any patches using that opatch.

Files not available to copy at the end of installation results in non working applications – WebLogic only

Symptom

If you choose the option **No. Configure but do not install the application** in the installer screen titled **Manual Deployment Option**, necessary wallet files that are required for application run time are deleted at the end of the installation.

Solution

Manual Deployment is not currently available in this installer. Choose **Yes. I have write access to the application server** in the installer screen, **Manual Deployment Option**.

> **Note:** To successfully perform this option, you also need to run the installer as a user with write access to the WebLogic installation.

F

Appendix: Database Parameter File

The database parameter file contains the initial settings to create and run an 11.1.0.7 database.

```
###############################################################################
# Oracle 11.1.0.x Parameter file
#
# NOTES: Before using this script:
#        1.  Change <datafile_path>, <admin_path>, <utl_file_path>, <diag_path>
and <hostname>
#            values as appropriate.
#        2.  Replace the word SID with the database name.
#        3.  Size parameters as necessary for development, test, and production
environments.
# ----------------------------------------------------------------------
# MAINTENANCE LOG
#
# Date     By          Parameter           Old/New         Notes
# +------+ +---------+ +-----------------+ +-------------+ +-------------+
#
#
###############################################################################
# ------------------------------------------------------------------------------
# The policy is to give 60% for sga and 40% for PGA out of Memory Target at
startup
# ------------------------------------------------------------------------------
memory_target                           = 2000M
# ------------------------------------------------------------------------------
audit_file_dest              = <admin_path>/adump
compatible               = 11.1.0
control_files                           = (<datafile_path>/control01.ctl
                                          ,<datafile_path>/control02.ctl)
db_block_size            = 8192      # Default is 2k; adjust before db creation,
cannot change after db is created
db_file_multiblock_read_count        = 16        # Platform specific (max io
size)/(block size)
db_name              = SID
diagnostic_dest                         = '<diag_path>'
java_pool_size           = 100M
job_queue_processes          = 5         # Oracle Retail required; number of
cpu's + 1
local_listener               =
"(ADDRESS=(PROTOCOL=TCP)(HOST=<hostname>)(PORT=1521))"
nls_calendar                            = GREGORIAN
nls_date_format              = DD-MON-RR # Oracle Retail required; if RDW
database see later entry for proper format
nls_language             = AMERICAN  # Default
nls_numeric_characters       = ".,"      # Should be explicitly set to ensure all
users/batch get the same results
nls_sort             = BINARY    # Should be explicitly set to ensure all
sessions get the same order
nls_territory            = AMERICA   # Default
open_cursors             = 900       # Oracle Retail required (minimum=900);
default is 50
optimizer_features_enable    = 11.1.0.7
```

```
optimizer_mode            = CHOOSE    # Oracle Retail required
plsql_optimize_level                  = 2         # 10g change; use this setting
to optimize plsql performance
processes                 = 2000       # Max number of OS  processes that can
connect to the db
query_rewrite_enabled         = TRUE      # Oracle Retail required for function-
based indexes
session_cached_cursors                = 900       # Oracle Retail required;
undo_management              = AUTO
undo_retention            = 1800      # Currently set for 30 minutes; set to avg
length of transactions in sec
undo_tablespace              = undo_ts
user_dump_dest            = <admin_path>/udump
utl_file_dir                          = <utl_file_path>
workarea_size_policy                  = auto          # Should be set to auto
when pga_aggregate_target is set
#
# ***  Set these parameters for Oracle Retail Data Warehouse (RDW) database ***
#nls_date_format              = DD-MON-RRRR  # Required by MicroStrategy
#query_rewrite_integrity               = TRUSTED
#star_transformation_enabled           = TRUE
#utl_file_dir                          = <Windows_utl_file_path>,
<UNIX_util_file_path>
#
# ***  Archive Logging, set if needed  ***
#log_archive_dest_1                    = 'location=<admin_path>/arch/'
#log_archive_format              = SIDarch_%r_%s_%t.log
#log_buffer           = 10485760    # Set to (512K or 128K)*CPUs
#log_checkpoint_interval               = 51200     # Default:0 - unlimited
#log_checkpoint_timeout                = 7200      # Default:1800 seconds
```

G

Appendix: Oracle Single Sign-On for Oracle Application Server (OAS)

Single Sign-On (SSO) is a term for the ability to sign onto multiple web applications via a single user ID/Password. There are many implementations of SSO – Oracle currently provides three different implementations: Oracle Single Sign-On (SSO), Java SSO (with the 10.1.3.1 release of OC4J) and Oracle Access Manager (provides more comprehensive user access capabilities).

Most, if not all, SSO technologies use a session cookie to hold encrypted data passed to each application. The SSO infrastructure has the responsibility to validate these cookies and, possibly, update this information. The user is directed to log on only if the cookie is not present or has become invalid. These session cookies are restricted to a single browser session and are never written to a file.

Another facet of SSO is how these technologies redirect a user's Web browser to various servlets. The SSO implementation determines when and where these redirects occur and what the final screen shown to the user is.

Most SSO implementations are performed in an application's infrastructure and not in the application logic itself. Applications that leverage infrastructure managed authentication (such as deploying specifying "Basic" or "Form" authentication) typically have little or no code changes when adapted to work in an SSO environment.

What Do I Need for Oracle Single Sign-On?

The nexus of an Oracle Single Sign-On system is the Oracle Identity Management Infrastructure installation. This consists of the following components:

- An Oracle Internet Directory (OID) LDAP server, used to store user, role, security, and other information. OID uses an Oracle database as the back-end storage of this information.
- An Oracle Single Sign-On servlet, used to authenticate the user and create the SSO session cookie. This servlet is deployed within the infrastructure Oracle Application Server (OAS).
- The Delegated Administration Services (DAS) application, used to administer users and group information. This information may also be loaded or modified via standard LDAP Data Interchange Format (LDIF) scripts.
- Additional administrative scripts for configuring the OSSO system and registering HTTP servers.

Additional OAS servers will be needed to deploy the business applications leveraging the OSSO technology.

Can Oracle Single Sign-On Work with Other SSO Implementations?

Yes, OSSO has the ability to interoperate with many other SSO implementations, but some restrictions exist.

Oracle Single Sign-on Terms and Definitions

Authentication

Authentication is the process of establishing a user's identity. There are many types of authentication. The most common authentication process involves a user ID and password.

Dynamically Protected URLs

A "Dynamically Protected URL" is a URL whose implementing application is aware of the OSSO environment. The application may allow a user limited access when the user has not been authenticated. Applications that implement dynamic OSSO protection typically display a "Login" link to provide user authentication and gain greater access to the application's resources.

Identity Management Infrastructure

The Identity Management Infrastructure is the collection of product and services which provide Oracle Single Sign-on functionality. This includes the Oracle Internet Directory, an Oracle HTTP server, and the Oracle Single Sign-On services. The Oracle Application Server deployed with these components is typically referred as the "Infrastructure" instance.

MOD_OSSO

mod_osso is an Apache Web Server module an Oracle HTTP Server uses to function as a partner application within an Oracle Single Sign-On environment. The Oracle HTTP Server is based on the Apache HTTP Server.

Oracle Internet Directory

Oracle Internet Directory (OID) is an LDAP-compliant directory service. It contains user ids, passwords, group membership, privileges, and other attributes for users who are authenticated using Oracle Single Sign-On.

Partner Application

A partner application is an application that delegates authentication to the Oracle Identity Management Infrastructure. One such partner application is the Oracle HTTP Server (OHS) supplied with the Oracle Application Server. OHS uses the MOD_OSSO module to configure this functionality.

All partner applications must be registered with the Oracle Single Sign-On server. An output product of this registration is a configuration file the partner application uses to verify a user has been previously authenticated.

Realm

A Realm is a collection users and groups (roles) managed by a single password policy. This policy controls what may be used for authentication (for example, passwords, X.509 certificates, and biometric devices). A Realm also contains an authorization policy used for controlling access to applications or resources used by one or more applications.

A single OID can contain multiple Realms. This feature can consolidate security for retailers with multiple banners or to consolidate security for multiple development and test environments.

Statically Protected URLs

A URL is considered to be "Statically Protected" when an Oracle HTTP server is configured to limit access to this URL to only OSSO authenticated users. Any attempt to access a "Statically Protected URL" results in the display of a login page or an error page to the user.

Servlets, static HTML pages, and JSP pages may be statically protected.

What Single Sign-On is not

Single Sign-On is NOT a user ID/password mapping technology.

However, some applications can store and retrieve user IDs and passwords for non-SSO applications within an OID LDAP server. An example of this is the Oracle Forms Web Application framework, which maps OSSO user IDs to a database logins on a per-application basis.

How Oracle Single Sign-On Works

Oracle Single Sign-On involves a couple of different components. These are:

- The Oracle Single Sign-On (OSSO) servlet, which is responsible for the back-end authentication of the user.
- The Oracle Internet Directory LDAP server, which stores user IDs, passwords, and group (role) membership.
- The Oracle HTTP Server associated with the web application, which verifies and controls browser redirection to the OSSO servlet.
- If the web application implements dynamic protection, then the web application itself is involved with the OSSO system.

Statically Protected URLs

When an unauthenticated user accesses a statically protected URL, the following occurs:

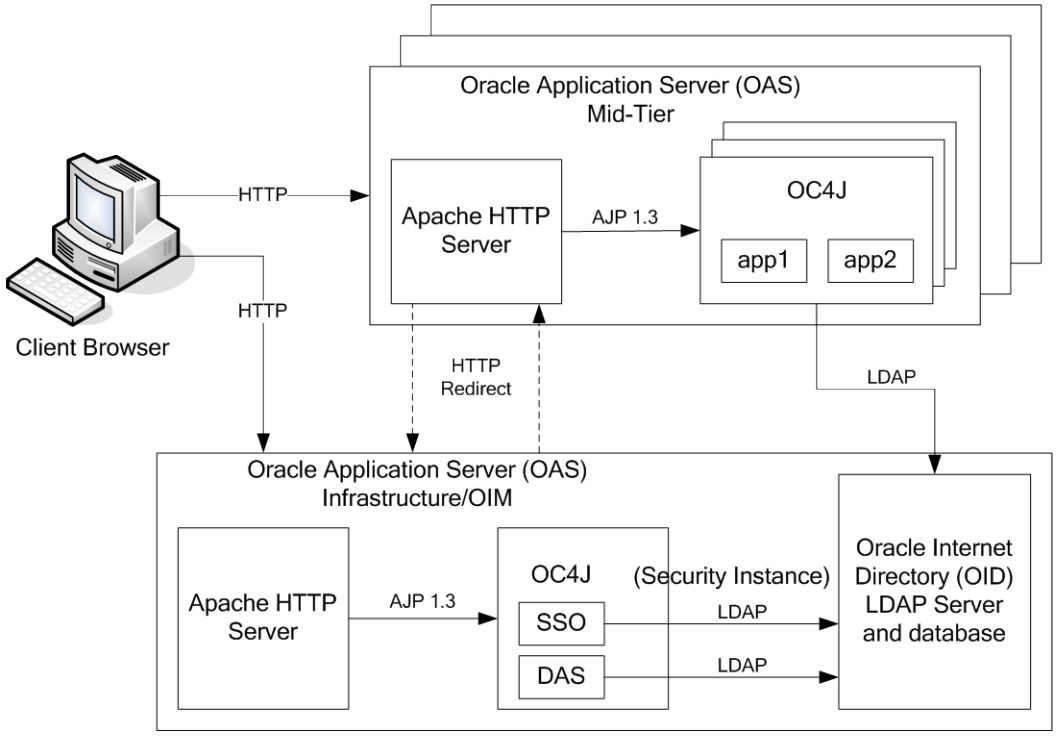
1. The Oracle HTTP server recognizes the user has not been authenticated and redirects the browser to the Oracle Single Sign-On servlet.
2. The OSSO servlet determines the user must authenticate, and displays the OSSO login page.
3. The user must sign in via a valid user ID and password. If the OSSO servlet has been configured to support multiple Realms, a valid realm must also be entered. The user ID, password, and realm information is validated against the Oracle Internet Directory LDAP server.
4. The OSSO servlet creates and sends the user's browser an OSSO session cookie. This cookie is never persisted to disk and is specific only to the current browser session. This cookie contains the user's authenticated identity. It does NOT contain the user's password.
5. The OSSO servlet redirects the user back to the Oracle HTTP Server, along with OSSO specific information.
6. The Oracle HTTP Server decodes the OSSO information, stores it with the user's session, and allows the user access to the original URL.

Dynamically Protected URLs

When an unauthenticated user accesses a dynamically protected URL, the following occurs:

1. The Oracle HTTP server recognizes the user has not been authenticated, but allows the user to access the URL.
2. The application determines the user must be authenticated and sends the Oracle HTTP server a specific status to begin the authentication process.
3. The Oracle HTTP Server redirects the user's browser session to the OSSO Servlet.
4. The OSSO servlet determines the user must authenticate, and displays the OSSO login page.
5. The user must sign in via a valid user ID and password. If the OSSO servlet has been configured to support multiple Realms, a valid realm must also be entered. The user ID, password, and realm information is validated against the Oracle Internet Directory LDAP server.
6. The OSSO servlet creates and sends the user's browser an OSSO session cookie. This cookie is never persisted to disk and is specific only to the current browser session. This cookie contains the user's authenticated identity. It does NOT contain the user's password.
7. The OSSO servlet redirects the user back to the Oracle HTTP Server, along with OSSO specific information.
8. The Oracle HTTP Server decodes the OSSO information, stores it with the user's session, and allows the user access to the original URL.

Single Sign-on Topology



Installation Overview

Installing Oracle Single Sign-On consists of installing the following components:

1. Installing the Oracle Internet Directory (OID) LDAP server and the Infrastructure Oracle Application Server (OAS). These are typically performed using a single session of the Oracle Universal Installer and are performed at the same time. OID requires an Oracle relational database and if one is not available, the installer will also install this as well.

   The Infrastructure OAS includes the Delegated Administration Services (DAS) application as well as the OSSO servlet. The DAS application can be used for user and realm management within OID.
2. Installing additional OAS 10.1.2 midtier instances for the Oracle Retail applications, such as RMS, that are based on Oracle Forms technologies. These instances must be registered with the Infrastructure OAS installed in step 1).
3. Installing additional application servers to deploy other Oracle Retail applications and performing application specific initialization and deployment activities.

Infrastructure Installation and Configuration

The Infrastructure installation for OSSO is dependent on the environment and requirements for its use. Deploying an Infrastructure OAS to be used in a test environment does not have the same availability requirements as for a production environment. Similarly, the Oracle Internet Directory (OID) LDAP server can be deployed in a variety of different configurations. See the *Oracle Application Server Installation Guide* and the *Oracle Internet Directory Installation Guide* for more details.

OID User Data

Oracle Internet Directory is an LDAP v3 compliant directory server. It provides standards-based user definitions out of the box.

The current version of Oracle Single Sign-On only supports OID as its user storage facility. Customers with existing corporate LDAP implementations may need to synchronize user information between their existing LDAP directory servers and OID. OID supports standard LDIF file formats and provides a JNDI compliant set of Java classes as well. Moreover, OID provides additional synchronization and replication facilities to integrate with other corporate LDAP implementations.

Each user ID stored in OID has a specific record containing user specific information. For role-based access, groups of users can be defined and managed within OID. Applications can thus grant access based on group (role) membership saving administration time and providing a more secure implementation.

OID with Multiple Realms

OID and OSSO can be configured to support multiple user Realms. Each realm is independent from each other and contains its own set of user IDs. As such, creating a new realm is an alternative to installing multiple OID and Infrastructure instances. Hence, a single Infrastructure OAS can be used to support many development and test environments by defining one realm for each environment.

Realms may also be used to support multiple groups of external users, such as those from partner companies. For more information on Realms, see the Oracle Internet Directory Administrators Guide.

User Management

User Management consists of displaying, creating, updating or removing user information. There are two basic methods of performing user management: LDIF scripts and the Delegate Administration Services (DAS) application.

OID DAS

The DAS application is a web based application designed for both administrators and users. A user may update their password, change their telephone number of record, or modify other user information. Users may search for other users based on partial strings of the user's name or ID. An administrator may create new users, unlock passwords, or delete users.

The DAS application is fully customizable. Administrators may define what user attributes are required, optional or even prompted for when a new user is created.

Furthermore, the DAS application is secure. Administrators may also what user attributes are displayed to other users. Administration is based on permission grants, so different users may have different capabilities for user management based on their roles within their organization.

LDIF Scripts

Script based user management can be used to synchronize data between multiple LDAP servers. The standard format for these scripts is the LDAP Data Interchange Format (LDIF). OID supports LDIF script for importing and exporting user information. LDIF scripts may also be used for bulk user load operations.

User Data Synchronization

The user store for Oracle Single Sign-On resides within the Oracle Internet Directory (OID) LDAP server. Oracle Retail applications may require additional information attached to a user name for application-specific purposes and may be stored in an application-specific database. Currently, there are no Oracle Retail tools for synchronizing changes in OID stored information with application-specific user stores. Implementers should plan appropriate time and resources for this process. Oracle Retail strongly suggests that you configure any Oracle Retail application using an LDAP for its user store to point to the same OID server used with Oracle Single Sign-On.

Setting up SIM for Single Sign-on

To set up Forms for Single Sign-on, the Forms framework must know and/or be configured to use SSO. To do this, the Forms framework configuration file formsweb.cfg must be configured to enable SSO and the mid-tier HTTP Server must be registered with the Oracle Single Sign-On server. In addition, the Forms framework uses Resource Access Descriptor (RAD), to map OSSO user IDs to Database connect strings.

Configuring formsweb.cfg

For each Forms application instance there are two attributes in the Forms framework configuration file `formsweb.cfg` that control SSO behavior:

Name	Value	Description
ssoMode	true/false	Enables/disables SSO
ssoDynamicResourceCreate	true/false	Enables/disables the dynamic RAD entry creation

Creating a RAD Entry

There are three ways by which a RAD entry (mapping an OSSO user ID to a Database connect string) may be created:

- Administrator Created

  An administrator uses the Delegated Administration Services (DAS) web application that comes with the infrastructure server and that can be launched using the URL http://*<host>:port/oiddas*.

- User Created

  The user can dynamically create a RAD entry when the Forms framework prompts the user for information. This however requires that the ssoDynamicResourceCreate attribute be set to true. If a RAD already exists, the user may also create additional RADs via the DAS application.

- LDIF Script

  More information about how to use an LDIF script to create a RAD entry may be found by accessing My Oracle Support document 244526.1.

H

Appendix: Oracle Single Sign-On for WebLogic

Single Sign-On (SSO) is a term for the ability to sign onto multiple Web applications via a single user ID/Password. There are many implementations of SSO – Oracle currently provides three different implementations: Oracle Single Sign-On (OSSO), Java SSO (with the 10.1.3.1 release of OC4J) and Oracle Access Manager (provides more comprehensive user access capabilities).

Most, if not all, SSO technologies use a session cookie to hold encrypted data passed to each application. The SSO infrastructure has the responsibility to validate these cookies and, possibly, update this information. The user is directed to log on only if the cookie is not present or has become invalid. These session cookies are restricted to a single browser session and are never written to a file.

Another facet of SSO is how these technologies redirect a user's Web browser to various servlets. The SSO implementation determines when and where these redirects occur and what the final screen shown to the user is.

Most SSO implementations are performed in an application's infrastructure and not in the application logic itself. Applications that leverage infrastructure managed authentication (such as deploying specifying "Basic" or "Form" authentication) typically have little or no code changes when adapted to work in an SSO environment.

What Do I Need for Oracle Single Sign-On?

The nexus of an Oracle Single Sign-On system is the Oracle Identity Management Infrastructure installation. This consists of the following components:

- An Oracle Internet Directory (OID) LDAP server, used to store user, role, security, and other information. OID uses an Oracle database as the back-end storage of this information.
- An Oracle HTTP Server 11g Release 1 as a front end to the Oracle WebLogic Server. The Oracle HTTP Server is included in the Oracle Web Tier Utilities 11g Release 1 (11.1.1).
- An Oracle Single Sign-On Plug-in, used to authenticate the user and create the OSSO session cookie. This is available in the Oracle Fusion Middleware 11g Web Tier Utilities (11.1.1.20) package.
- The Delegated Administration Services (DAS) application, used to administer users and group information. This information may also be loaded or modified via standard LDAP Data Interchange Format (LDIF) scripts.
- Additional administrative scripts for configuring the OSSO system and registering HTTP servers.

For more information on setting up single sign on, refer to the *Oracle Retail Predictive Application Server Installation Guide*.

Additional WebLogic managed servers will be needed to deploy the business applications leveraging the OSSO technology.

Can Oracle Single Sign-On Work with Other SSO Implementations?

Yes, OSSO has the ability to interoperate with many other SSO implementations, but some restrictions exist.

Oracle Single Sign-on Terms and Definitions

Authentication

Authentication is the process of establishing a user's identity. There are many types of authentication. The most common authentication process involves a user ID and password.

Dynamically Protected URLs

A "Dynamically Protected URL" is a URL whose implementing application is aware of the OSSO environment. The application may allow a user limited access when the user has not been authenticated. Applications that implement dynamic OSSO protection typically display a "Login" link to provide user authentication and gain greater access to the application's resources.

Identity Management Infrastructure

The Identity Management Infrastructure is the collection of product and services which provide Oracle Single Sign-on functionality. This includes the Oracle Internet Directory, an Oracle HTTP server, and the Oracle Single Sign-On services. The Oracle Application Server deployed with these components is typically referred as the Infrastructure instance.

MOD_OSSO

mod_osso is an Apache Web Server module an Oracle HTTP Server uses to function as a partner application within an Oracle Single Sign-On environment. The Oracle HTTP Server is based on the Apache HTTP Server.

MOD_WEBLOGIC

mod_weblogic operates as a module within the HTTP server that allows requests to be proxied from the Apache HTTP server to the WebLogic server.

Oracle Internet Directory

Oracle Internet Directory (OID) is an LDAP-compliant directory service. It contains user ids, passwords, group membership, privileges, and other attributes for users who are authenticated using Oracle Single Sign-On.

Partner Application

A partner application is an application that delegates authentication to the Oracle Identity Management Infrastructure. One such partner application is the Oracle HTTP Server (OHS) supplied with the Oracle Application Server. OHS uses the MOD_OSSO module to configure this functionality.

All partner applications must be registered with the Oracle Single Sign-On server. An output product of this registration is a configuration file the partner application uses to verify a user has been previously authenticated.

Realm

A Realm is a collection users and groups (roles) managed by a single password policy. This policy controls what may be used for authentication (for example, passwords, X.509 certificates, and biometric devices). A Realm also contains an authorization policy used for controlling access to applications or resources used by one or more applications.

A single OID can contain multiple Realms. This feature can consolidate security for retailers with multiple banners or to consolidate security for multiple development and test environments.

Statically Protected URLs

A URL is considered to be "Statically Protected" when an Oracle HTTP server is configured to limit access to this URL to only SSO authenticated users. Any attempt to access a "Statically Protected URL" results in the display of a login page or an error page to the user.

Servlets, static HTML pages, and JSP pages may be statically protected.

> **Note:** Dynamically Protected URL and Statically Protected URL are within the context of the Oracle Software Security Assurance (OSSA). The static protection for URLs is a common JEE feature.

What Single Sign-On is not

Single Sign-On is NOT a user ID/password mapping technology.

However, some applications can store and retrieve user IDs and passwords for non-SSO applications within an OID LDAP server. An example of this is the Oracle Forms Web Application framework, which maps OSSO user IDs to a database logins on a per-application basis.

How Oracle Single Sign-On Works

Oracle Single Sign-On involves a couple of different components. These are:

- The Oracle Single Sign-On (OSSO) servlet, which is responsible for the back-end authentication of the user.
- The Oracle Internet Directory LDAP server, which stores user IDs, passwords, and group (role) membership.
- The Oracle HTTP Server associated with the Web application, which verifies and controls browser redirection to the OSSO servlet.
- If the Web application implements dynamic protection, then the Web application itself is involved with the OSSO system.

Statically Protected URLs

When an unauthenticated user accesses a statically protected URL, the following occurs:

1. The user's Web browser makes an HTTP request to a protected URL serviced by the Oracle HTTP Server (OHS).
2. The Oracle HTTP Server processses the request and routes it to the mod_oss module.

3. This module determines whether the user is already authenticated. If the authentication is required, it directs the browser to the OSSO server. The OSSO server checks for a secure cookie containing the authentication information. If the cookie is not found, the following occurs:
   a. The OSSO servlet determines the user must authenticate, and displays the OSSO login page.
   b. The user must sign in via a valid user ID and password. If the OSSO servlet has been configured to support multiple Realms, a valid realm must also be entered. The user ID, password, and realm information is validated against the Oracle Internet Directory LDAP server. The browser is then redirected back to the Oracle HTTP Server with the encrypted authentication credentials. It does NOT contain the user's password.
4. The mod_osso module then decrypts the user credentials and sets HTTP headers with relevant user attributes, marking the user's session as authenticated.
5. The mod_weblogic module (within the Oracle HTTP Server) then forwards the request to the Oracle WebLogic Server.
6. The Oracle WebLogic Server then invokes the configured authentication providers that decode the headers and provide the user's role membership. In an OSSO implementation, ensure that the OSSO Identity Asserter is invoked and Oracle Internet Directory (OID) Authenticator is executed to provide the user's role membership.
7. Once the authentication is established, the relevant application logic is initiated and the response is sent back to the user through the Oracle HTTP Server.
   Because the Web browser session is now authenticated, subsequent requests in that session are not redirected to the OSSO server for authentication.

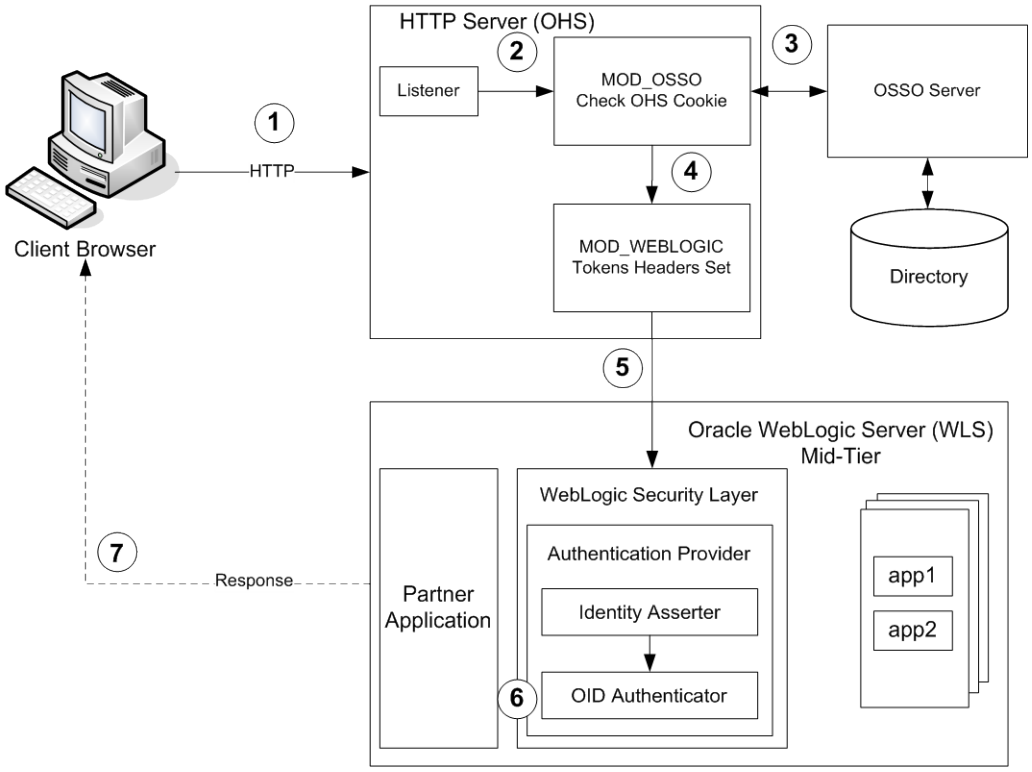
Dynamically Protected URLs

When an unauthenticated user accesses a dynamically protected URL, the following occurs:

1. The user's Web browser makes an HTTP request to a protected URL serviced by the Oracle HTTP Server (OHS). The Oracle HTTP server recognizes the user has not been authenticated, but allows the user to access the URL.
2. The application determines the user must be authenticated and send the Oracle HTTP Server a specific status to begin the authentication process.
3. The Oracle HTTP Server processses the request and routes it to the mod_oss module.
4. This module determines whether the user is already authenticated. If the authentication is required, it directs the browser to the OSSO server. The OSSO server checks for a secure cookie containing the authentication information. If the cookie is not found, the following occurs:
   a. The OSSO servlet determines the user must authenticate, and displays the OSSO login page.
   b. The user must sign in via a valid user ID and password. If the OSSO servlet has been configured to support multiple Realms, a valid realm must also be entered. The user ID, password, and realm information is validated against the Oracle Internet Directory LDAP server. The browser is then redirected back to the Oracle HTTP Server with the encrypted authentication credentials. It does NOT contain the user's password.
5. The mod_osso module then decrypts the user credentials and sets HTTP headers with relevant user attributes, marking the user's session as authenticated.

6. The mod_weblogic module (within the Oracle HTTP Server) then forwards the request to the Oracle WebLogic Server.
7. The Oracle WebLogic Server then invokes the configured authentication providers that decode the headers and provide the user's role membership. In an OSSO implementation, ensure that the OSSO Identity Asserter is invoked and Oracle Internet Directory (OID) Authenticator is executed to provide the user's role membership.
8. Once the authentication is established, the relevant application logic is initiated and the response is sent back to the user through the Oracle HTTP Server.
   Because the Web browser session is now authenticated, subsequent requests in that session are not redirected to the OSSO server for authentication.

Single Sign-on Topology



Installation Overview

Installing Oracle Single Sign-On consists of installing the following components:

1. Installing the Oracle Internet Directory (OID) LDAP server and the Infrastructure Oracle Application Server (OAS). These are typically performed using a single session of the Oracle Universal Installer and are performed at the same time. OID requires an Oracle relational database and if one is not available, the installer will also install this as well.

   The Infrastructure OAS includes the Delegated Administration Services (DAS) application as well as the OSSO servlet. The DAS application can be used for user and realm management within OID.

2. Installing additional midtier instances (such as OAS 10.1.4) for the Oracle Retail applications, such as RMS, that are based on Oracle Forms technologies. These instances must be registered with the Infrastructure OAS installed in step 1.
3. Installing additional application servers to deploy other Oracle Retail applications and performing application specific initialization and deployment activities.

Infrastructure Installation and Configuration

The Infrastructure installation for OSSO is dependent on the environment and requirements for its use. Deploying an Infrastructure OAS to be used in a test environment does not have the same availability requirements as for a production environment. Similarly, the Oracle Internet Directory (OID) LDAP server can be deployed in a variety of different configurations. See the *Oracle Application Server Installation Guide* and the *Oracle Internet Directory Installation Guide* for more details.

OID User Data

Oracle Internet Directory is an LDAP v3 compliant directory server. It provides standards-based user definitions out of the box.

The current version of Oracle Single Sign-On only supports OID as its user storage facility. Customers with existing corporate LDAP implementations may need to synchronize user information between their existing LDAP directory servers and OID. OID supports standard LDIF file formats and provides a JNDI compliant set of Java classes as well. Moreover, OID provides additional synchronization and replication facilities to integrate with other corporate LDAP implementations.

Each user ID stored in OID has a specific record containing user specific information. For role-based access, groups of users can be defined and managed within OID. Applications can thus grant access based on group (role) membership saving administration time and providing a more secure implementation.

OID with Multiple Realms

OID and OSSO can be configured to support multiple user Realms. Each realm is independent from each other and contains its own set of user IDs. As such, creating a new realm is an alternative to installing multiple OID and Infrastructure instances. Hence, a single Infrastructure OAS can be used to support development and test environments by defining one realm for each environment.

Realms may also be used to support multiple groups of external users, such as those from partner companies. For more information on Realms, see the *Oracle Internet Directory Administrators Guide*.

User Management

User Management consists of displaying, creating, updating or removing user information. There are two basic methods of performing user management: LDIF scripts and the Delegate Administration Services (DAS) application.

OID DAS

The DAS application is a Web-based application designed for both administrators and users. A user may update their password, change their telephone number of record, or modify other user information. Users may search for other users based on partial strings of the user's name or ID. An administrator may create new users, unlock passwords, or delete users.

The DAS application is fully customizable. Administrators may define what user attributes are required, optional or even prompted for when a new user is created.

Furthermore, the DAS application is secure. Administrators may also what user attributes are displayed to other users. Administration is based on permission grants, so different users may have different capabilities for user management based on their roles within their organization.

LDIF Scripts

Script based user management can be used to synchronize data between multiple LDAP servers. The standard format for these scripts is the LDAP Data Interchange Format (LDIF). OID supports LDIF script for importing and exporting user information. LDIF scripts may also be used for bulk user load operations.

User Data Synchronization

The user store for Oracle Single Sign-On resides within the Oracle Internet Directory (OID) LDAP server. Oracle Retail applications may require additional information attached to a user name for application-specific purposes and may be stored in an application-specific database. Currently, there are no Oracle Retail tools for synchronizing changes in OID stored information with application-specific user stores. Implementers should plan appropriate time and resources for this process. Oracle Retail strongly suggests that you configure any Oracle Retail application using an LDAP for its user store to point to the same OID server used with Oracle Single Sign-On.

How to Configure Single Sign On Functionality on SIM

This is a step by step procedure to set SSO on SIM Oracle Retail Application for the present release. Follow the instructions contained in the FIRST and SECOND parts of this document

Part I

To get SSO functionality on SIM, there are some manual steps to adjust one of the properties files, the JnlpLaunch.properties file. When SIM was installed on the server, the JnlpLaunch.properties file was configured to work with the SIM managed server port number from your WebLogic installation. When SIM was deployed, WebLogic put two EAR files in the stage directory of the SIM managed server. The JnlpLaunch.properties file resides in these EAR files.

To get SSO working, replace the original port number for the Web Tier (OHS) port number in the JnlpLaunch.properties file.

To to replace the original managed server port number with the Web Tier OHS port number in the JnlpLaunch.properties file, complete the following steps.

1. Stop the SIM managed server.
2. JnlpLaunch.properties file is located in two of the EAR files that were deployed to your WebLogic installation and they are located here.

   ```
   $WEBLOGIC_ DOMAIN_HOME/servers/sim-server/stage/sim-client/sim-client.ear
   $WEBLOGIC_ DOMAIN_HOME/servers/sim-server/stage/sim-server/sim-server.ear
   ```

3. Make backups of both EAR files:
   - cp sim-client.ear sim-client.ear_BK
   - cp sim-server.ear sim-server.ear_BK

4. Create a /tmp-client directory and move sim-client.ear to this directory, expand the EAR file and modify JnlpLaunch.properties file.
   a. jar xvf sim-client.ear

      JnlpLaunch.properties file is inside ./conf directory
   b. cd to ./conf directory and modify JnlpLaunch.properties

      Example:

      ```
      Original value:
      "...
      ...
      token.sim_download_url=http://myserver.com:8002/sim-client
      ...
      ..."
      ```

      In this case, 8002 is the SIM managed Server port number.

      ```
      New value:
      "...
      token.sim_download_url=http://myserver.com:8888/sim-client
      ...
      ..."
      ```

      In this case, 8888 is the WebTier port number.

      To find the OHS port number in your Web Tier installation, enter from the command line:

      $ORACLE_WEB_TIER/instances/instance1/bin/opmnctl status -l

      Processes in Instance: instance1

      ```
      ---------------------------------------------------------------------------------------------------
      ias-component | process-type | ports
      ---------------------------------------------------------------------------------------------------
      ohs1 | OHS | https:8889,https:4443,http:8888
      ```

      The OHS port number is highlighted on the second line **http:8888**

      Do not change any other parameters and save JnlpLaunch.properties file.
5. cd to /tmp-client directory and compress the contents of this directory. Enter this command to keep all the entries in the META-INF/MANIFEST.MF file and to get the new sim-client.ear file:

   jar cvMf sim-client.ear .

   NOTE: This command must be typed in this way (jar cvMf) to keep all entries in the MANIFEST.MF FILE)
6. Move the new sim-client.ear file to its original location, in $WEBLOGIC_DOMAIN_HOME/servers/sim-server/stage/sim-client directory.
7. Create a /tmp-server directory and move sim-server.ear to this directory, expand the EAR file, modify JnlpLaunch.properties and compress directory contents to get new sim-server.ear.
   a. jar xvf sim-server.ear

      JnlpLaunch.properties file is in the root directory
   b. modify JnlpLaunch.properties file

      Example:

      ```
      Original value:
        "...
      ...
      token.sim_download_url=http://myserver.com:8002/sim-client
      ```

```
 ...
..."
 New value:
 "...
 ...
 token.sim_download_url=http://myserver.com:8888/sim-client
 ...
 ..."
```

Do not change any other parameter in this file and save JnlpLaunch.properties file.

c. From the root of /tmp-server directory compress the contents of the directory to get the new sim-server.ear file by entering this command:

jar cvMf sim-server.ear

> **Note:** This command must be typed in this way (jar cvMf) to keep all entries in the MANIFEST.MF FILE

d. Move the new sim-server.ear file to its original location, in `<WEBLOGIC_DOMAIN_HOME>/servers/sim-server/stage/<sim-server> directory`

e. Start the SIM managed server to update in WebLogic server the changes to your SIM files.

> **Note:** For SSO, SIM managed server should be started using the following parameter. Dweblogic.http.enableRemoteUserHeader=true.
>
> For example: <WEBLOGIC_DOMAIN_HOME>/bin/startManagedWeblogic.sh <sim-server> <server:port> -Dweblogic.http.enableRemoteUserHeader=true.
>
> If the <sim-server> is being started from WebLogic Admin console, the above parameter should be added here before restarting the SIM managed server from the Admin console: Adminconsole > Environment > Servers > <sim-server> > Server Start > Arguments.

Part II

1. When SIM is installed for SSO functionality, before launching SIM, switch the SIM database from INTERNAL SECURITY ACCESS to EXTERNAL SECURITY ACCESS Protect the application as follows:
   - **a.** Switch SIM database security access.
   - **b.** On the table RK_CONFIG', the column CONFIG_VALUE must be equal to 1 when the column CONFIG_KEY is equal to SECURITY_AUTHENTICATION_METHOD.

```
sql> UPDATE RK_CONFIG
    SET CONFIG_VALUE = '1'
    WHERE CONFIG_KEY = 'SECURITY_AUTHENTICATION_METHOD';
sql> commit;
```

2. In Oracle Web Tier (OHS), add some entries to two Oracle Web Tier files, mod_osso.conf and mod_wl_ohs.conf, as follows:
   - **a.** Enter two <Location> entries in the file mod_osso.conf located here:

     $ORACLE_WEB_TIER/instances/instance1/config/OHS/ohs1/moduleconf

```
"...
<Location /sim-client>
    WebLogicHost   myhostname.com
    WebLogicPort   SIM managed server port number
 </Location>
<Location /sim-client/launch>
    WebLogicHost   myhostname.com
    WebLogicPort   SIM managed server port number
    require valid-user
    AuthType Osso
 </Location>
 ..."
```

A real life example is as follows:

```
"...
 <Location /sim-client>
    WebLogicHost   myhostname.com
    WebLogicPort   8002
 </Location>
 <Location /sim-client/launch>
    WebLogicHost   myhostname.com
    WebLogicPort   8002
    require valid-user
    AuthType Osso
 </Location>
..."
```

b. Enter one entry for <Location> in the file mod_wl_ohs.conf located here:

$ORACLE_WEB_TIER/ instances/instance1/config/OHS/ohs1

```
 "...
    <Location /sim-client>
        SetHandler weblogic-handler
    </Location>
 ..."
```

c. Re-start OHS server to read the configuration changes.

$ORACLE_WEB_TIER/instances/instance1/bin/opmnctl stopproc ias-component=ohs1

$ORACLE_WEB_TIER/instances/instance1/bin/opmnctl startproc ias-component=ohs1

d. Launch SIM.

The SIM URL should include the OHS port number. To find the OHS port number in your Web Tier installation, enter from the command line:

$ORACLE_WEB_TIER/instances/instance1/bin/opmnctl status -l

Processes in Instance: instance1

```
-------------------------------------------------------------------------------------------------------------------
ias-component | process-type | ports
-------------------------------------------------------------------------------------------------------------------
ohs1          | OHS          | https:8889,https:4443,http:8888
```

The OHS port number is highlighted on the second line http:8888. The SIM URL should look like this:

http://myhostname.com:8888/sim-client/launch?template=sim_jnlp_template.vm

I

Appendix: Setting Up Password Stores with Oracle Wallet

As part of an application installation, administrators must set up password stores for database user accounts using Oracle Wallet. These password stores must be installed on the application database side. While the installer handles much of this process, the administrators must perform some additional steps.

A password store for the application and application server user accounts must also be installed; however, the installer takes care of this entire process.

About Password Stores and Oracle Wallet

Oracle databases have allowed other users on the server to see passwords in case database connect strings (username/password@db) were passed to programs. In the past, users could navigate to `ps -ef|grep <username>` to see the password if the password was supplied in the command line when calling a program.

To make passwords more secure, Oracle Retail has implemented the Oracle Software Security Assurance (OSSA) program. Sensitive information such as user credentials now must be encrypted and stored in a secure location. This location is called password stores or wallets. These password stores are secure software containers that store the encrypted user credentials.

Users can retrieve the credentials using aliases that were set up when encrypting and storing the user credentials in the password store. For example, if `username/password@db` is entered in the command line argument and the alias is called `db_username`, the argument to a program is as follows:

```
sqlplus /@db_username
```

This would connect to the database as it did previously, but it would hide the password from any system user.

After this is configured, as in the example above, the application installation and the other relevant scripts are no longer needed to use embedded usernames and passwords. This reduces any security risks that may exist because usernames and passwords are no longer exposed.

When the installation starts, all the necessary user credentials are retrieved from the Oracle Wallet based on the alias name associated with the user credentials.

There are two different types of password stores or wallets. One type is for database connect strings used in program arguments (such as `sqlplus /@db_username`). The other type is for Java application installation and application use.

Setting Up Password Stores for Database User Accounts

After the database is installed and the default database user accounts are set up, administrators must set up a password store using the Oracle wallet. This involves assigning an alias for the username and associated password for each database user account. The alias is used later during the application installation. This password store must be created on the system where the application server and database client are installed.

This section describes the steps you must take to set up a wallet and the aliases for the database user accounts. For more information on configuring authentication and password stores, see the *Oracle Database Security Guide*.

> **Note:** In this section, <wallet_location> is a placeholder text for illustration purposes. Before running the command, ensure that you specify the path to the location where you want to create and store the wallet.

To set up a password store for the database user accounts, perform the following steps:

3. Create a wallet using the following command:

   ```
   kstore -wrl <wallet_location> -create
   ```

   After you run the command, a prompt appears. Enter a password for the Oracle Wallet in the prompt.

   > **Note:** The mkstore utility is included in the Oracle Database Client installation.

   The wallet is created with the auto-login feature enabled. This feature enables the database client to access the wallet contents without using the password. For more information, refer to the *Oracle Database Advanced Security Administrator's Guide.*

4. Create the database connection credentials in the wallet using the following command:

   ```
   mkstore -wrl <wallet_location> -createCredential <alias-name> <database-user-name>
   ```

   After you run the command, a prompt appears. Enter the password associated with the database user account in the prompt.

5. Repeat Step 2 for all the database user accounts.

6. Update the sqlnet.ora file to include the following statements:

   ```
   WALLET_LOCATION = (SOURCE = (METHOD = FILE) (METHOD_DATA = (DIRECTORY = <wallet_location>)))
   SQLNET.WALLET_OVERRIDE = TRUE
   SSL_CLIENT_AUTHENTICATION = FALSE
   ```

7. Update the tnsnames.ora file to include the following entry for each alias name to be set up.

   ```
   <alias-name> =
       (DESCRIPTION =
        (ADDRESS_LIST =
               (ADDRESS = (PROTOCOL = TCP) (HOST = <host>) (PORT = <port>))
           )
           (CONNECT_DATA =
               (SERVICE_NAME = <service>)
            )
    )
   ```

In the previous example, <alias-name>, <host>, <port>, and <service> are placeholder text for illustration purposes. Ensure that you replace these with the relevant values.

Setting Up Wallets for Database User Accounts

The following examples show how to set up wallets for database user accounts for the following applications:

- For RMS, RWMS, RPM Batch, RETL, RMS, RWMS, and ARI
- For Java Applications (SIM, ReIM, RPM, Alloc, RIB, RSL, AIP, RETL)

For RMS, RWMS, RPM Batch, RETL, RMS, RWMS, and ARI

1. Create a new directory called wallet under your folder structure.

```
cd /projects/rms13.2/dev/
mkdir .wallet
```

> **Note:** The default permissions of the wallet allow only the owner to use it, ensuring the connection information is protected. If you want other users to be able to use the connection, you must adjust permissions appropriately to ensure only authorized users have access to the wallet.

2. Create a sqlnet.ora in the wallet directory with the following content.

```
WALLET_LOCATION =   (SOURCE =     (METHOD = FILE)     (METHOD_DATA =
(DIRECTORY =  /projects/rms13.2/dev/.wallet)) )
SQLNET.WALLET_OVERRIDE=TRUE
SSL_CLIENT_AUTHENTICATION=FALSE
```

> **Note**: WALLET_LOCATION must be on line 1 in the file.

3. Setup a tnsnames.ora in the wallet directory. This tnsnames.ora includes the standard tnsnames.ora file. Then, add two custom tns_alias entries that are only for use with the wallet. For example, `sqlplus /@dvols29_rms01user`.

```
ifile = /u00/oracle/product/11.2.0.1/network/admin/tnsnames.ora

dvols29_rms01user =
  (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = tcp)
  (host = mspdv311.us.oracle.com) (Port = 1521)))
    (CONNECT_DATA = (SID = dvols29) (GLOBAL_NAME = dvols29)))

dvols29_rms01user.world =
  (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = tcp)
  (host = mspdv311.us.oracle.com) (Port = 1521)))
    (CONNECT_DATA = (SID = dvols29) (GLOBAL_NAME = dvols29)))
```

> **Note**: It is important to not just copy the tnsnames.ora file because it can quickly become out of date. The ifile clause (shown above) is key.

4. Create the wallet files. These are empty initially.
   a. Ensure you are in the intended location.
      ```
      $ pwd
      /projects/rms13.2/dev/.wallet
      ```
   b. Create the wallet files.
      ```
      $ mkstore -wrl . –create
      ```
   c. Enter the wallet password you want to use. It is recommended that you use the same password as the UNIX user you are creating the wallet on.
   d. Enter the password again.

      Two wallet files are created from the above command:
      - ewallet.p12
      - cwallet.sso
5. Create the wallet entry that associates the user name and password to the custom tns alias that was setup in the wallet's tnsnames.ora file.

   ```
   mkstore –wrl . –createCredential <tns_alias> <username> <password>
   ```

   Example: `mkstore –wrl . –createCredential dvols29_rms01user rms01user passwd`
6. Test the connectivity. The ORACLE_HOME used with the wallet must be the same version or higher than what the wallet was created with.

   ```
   $ export TNS_ADMIN=/projects/rms13.2/dev/.wallet /* This is very import to use
   wallet to point at the alternate tnsnames.ora created in this example */

   $ sqlplus /@dvols29_rms01user

   SQL*Plus: Release 11

   Connected to:
   Oracle Database 11g

   SQL> show user
   USER is “rms01user”
   ```

   Running batch programs or shell scripts would be similar:

   ```
   Ex: dtesys /@dvols29_rms01user
   script.sh /@dvols29_rms01user

   Set the UP unix variable to help with some compiles :

   export UP=/@dvols29_rms01user
   for use in RMS batch compiles, and RMS, RWMS, and ARI forms compiles.
   ```

   As shown in the example above, users can ensure that passwords remain invisible.

Additional Database Wallet Commands

The following is a list of additional database wallet commands.

- Delete a credential on wallet

  ```
  mkstore –wrl . –deleteCredential dvols29_rms01user
  ```
- Change the password for a credential on wallet

  ```
  mkstore –wrl . –modifyCredential dvols29_rms01user rms01user passwd
  ```

- List the wallet credential entries

  ```
  mkstore –wrl . –list
  ```

  This command returns values such as the following.

  ```
  oracle.security.client.connect_string1
  oracle.security.client.user1
  oracle.security.client.password1
  ```

- View the details of a wallet entry

  ```
  mkstore –wrl . –viewEntry oracle.security.client.connect_string1
  ```

  Returns the value of the entry:

  ```
  dvols29_rms01user
  mkstore –wrl . –viewEntry oracle.security.client.user1
  ```

  Returns value of the entry:

  ```
  rms01user
  ```

  ```
  mkstore –wrl . –viewEntry oracle.security.client.password1
  ```

  Returns value of the entry:

  ```
  passwd
  ```

For Java Applications (SIM, ReIM, RPM, Alloc, RIB, RSL, AIP, RETL)

For Java application, consider the following:

- For database user accounts, ensure that you set up the same alias names between the password stores (database wallet and Java wallet). You can provide the alias name during the installer process.
- Document all aliases that you have set up. During the application installation, you must enter the alias names for the application installer to connect to the database and application server.
- Java wallets do not have a password to update their entries. Entries in Java wallets are stored in partitions, or application-level keys. In each retail application, after you unzip <app>13application.zip, cd into <app>/application/retail-public-security-api/bin (for example, mspdv351:[1033_WLS] /u00/webadmin/product/10.3.3/WLS/user_projects/domains/132_mck_soa_domain/reim13/wallet/bin

  or

  Unzip to

  reim/application/retail-public-security-api-bin to run the commands below to adminster java wallets. The application installers should create the Java wallets for you, but it is good to know how this works for future use and understanding.

  There are two scripts relating to this at that folder: dump_credentials.sh and save_credential.sh.

dump_credentials.sh

dump_credentials.sh is used to retrieve information from wallet. For each entry found in the wallet, the wallet partition, the alias, and the user name are displayed. Note that the password is not displayed. If the value of an entry is uncertain, run save_credential.sh to resave the entry with a known password.

```
dump_credentials.sh <wallet location>
```

Example:

```
dump_credentials.sh
location:/u00/webadmin/product/10.3.3/WLS/user_projects/domains/132_mck_soa_do
main/sim-client/csm

Retail Public Security API Utility
==========================================
```

Below are the credentials found in the wallet at the location:/u00/webadmin/product/10.3.3/WLS/user_projects/domains/132_mck_soa_domain/retail/reim13/config

```
==========================================

Apapplication level key partition name:reim13
User Name Alias:WLS-ALIAS User Name:weblogic
User Name Alias:RETAIL-ALIAS User Name:retail.user
User Name Alias:LDAP-ALIAS User Name:RETAIL.USER
User Name Alias:RMS-ALIAS User Name:rms132mock
User Name Alias:REIMBAT-ALIAS User Name:reimbat
```

save_credential.sh

save_credential.sh is used to update the information in wallet. If you are unsure about the information that is currently in the wallet, use dump_credentials.sh as indicated above.

```
save_credential.sh -a <alias> -u <user> -p <partition name>  –l <path of the
wallet file location where credentials are stored>
```

Example:

```
mspdv351:[1033_WLS]
/u00/webadmin/mock132_testing/rtil/rtil/application/retail-public-security-
api/bin> save_credential.sh -l wallet_test -a myalias -p mypartition -u myuser

==========================================
Retail Public Security API Utility
==========================================

Enter password:
Verify password:
```

Note: -p in the above command is for partition name. You must specify the proper partition name used in application code for each Java application.

save_credential.sh and dump_credentials.sh scripts are the same for all applications. If using save_credential.sh to add a wallet entry or to update a wallet entry, bounce the application/managed server so that your changes are visible to the application. Also, save a backup copy of your cwallet.sso file in a location outside of the deployment path, because redeployment or reinstallation of the application will wipe the wallet entries you made after installation of the application. To restore your wallet entries after a redeployment/reinstallation, copy the backed up cwallet.sso file over the cwallet.sso file. Then bounce the application/managed server.

Usage

```
==========================================
Retail Public Security API Utility
==========================================
usage: save_credential.sh -au[plh]
E.g. save_credential.sh -a rms-alias -u rms_user -p rib-rms -l ./
 -a,--userNameAlias <arg>              alias for which the credentials
needs to be stored
 -h,--help                             usage information
 -l,--locationofWalletDir <arg>        location where the wallet file is
created.If not specified, it creates the wallet under secure-credential-wallet
directory which is already present under the retail-public-security-api/
directory.
 -p,--appLevelKeyPartitionName <arg>   application level key partition name
 -u,--userName <arg>                   username to be stored in secure
credential wallet for specified alias*
```

How does the Wallet tie back to the Application?

The ORACLE Retail Java applications have the wallet alias information you create in an <app-name>.properties file. Below is the reim.properties file. Note the database information and the user are presented as well. The property called datasource.credential.alias=RMS-ALIAS uses the ORACLE wallet with the argument of RMS-ALIAS at the csm.wallet.path and csm.wallet.partition.name = reim13 to retrieve the password for application use.

Reim.properties code sample:

```
datasource.url=jdbc:oracle:thin:@mspdv349.us.oracle.com:1521:pkols07
datasource.schema.owner=rms132mock
datasource.credential.alias=RMS-ALIAS
# ===============================================================
# ossa related Configuration
#
# These settings are for ossa configuration to store credentials.
# ===============================================================

csm.wallet.path=/u00/webadmin/product/10.3.3/WLS/user_projects/domains/132_mck_soa
_domain/retail/reim13/config
csm.wallet.partition.name=reim
```

How does the Wallet tie back to java pgm batch use (such as ReIM batch)?

Some of the ORACLE Retail Java batch applications have an alias to use when running Java batch programs. For example, alias REIMBAT-ALIAS maps through the wallet to dbuser reimbat, already on the database. To run a ReIM batch program the format would be: reimbatchpgmname REIMBAT-ALIAS <other arguments as needed by the program in question>

Setting up RETL Wallets for 13.2

RETL 13.2 creates a wallet under $RFX_HOME/etc/security, with the following files:

- cwallet.sso
- jazn-data.xml
- jps-config.xml
- README.txt

To set up RETL wallets, perform the following steps:

1. Set the following environment variables:
   - `ORACLE_SID=<retaildb>`
   - `RFX_HOME=/u00/rfx/rfx-13.2.0`
   - `RFX_TMP=/u00/rfx/rfx-13.2.0/tmp`
   - `JAVA_HOME=/usr/jdk1.6.0_12.64bit`
   - `LD_LIBRARY_PATH=$ORACLE_HOME`
   - `PATH=$RFX_HOME/bin:$JAVA_HOME/bin:$PATH`
2. Change directory to `$RFX_HOME/bin`.
3. Run `setup-security-credential.sh`.
   - Enter 1 to add a new database credential.
   - Enter the dbuseralias. For example, `retl_java_rms01user`.
   - Enter the database user name. For example, `rms01user.`
   - Enter the database password.
   - Re-enter the database password.
   - Enter D to exit the setup script.
4. Update your RETL environment variable script to reflect the names of both the Oracle Networking wallet and the Java wallet.

   For example, to configure RETLforRPAS, modify the following entries in `$MMHOME/RETLforRPAS/rfx/etc/rmse_rpas_config.env.`
   - The RETL_WALLET_ALIAS should point to the Java wallet entry:
     `export RETL_WALLET_ALIAS="retl_java_rms01user"`
   - The ORACLE_WALLET_ALIAS should point to the Oracle network wallet entry:
     `export ORACLE_WALLET_ALIAS="dvols29_rms01user"`
   - The SQLPLUS_LOGON should use the ORACLE_WALLET_ALIAS:
     `export SQLPLUS_LOGON="/@${ORACLE_WALLET_ALIAS}"`
5. To change a password later, run `setup-security-credential.sh`.
   - Enter 2 to update a database credential.
   - Select the credential to update.
   - Enter the database user to update or change.
   - Enter the password of the database user.
   - Re-enter the password.

Quick Guide for Retail Wallets

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
RMS batch	DB	<RMS batch install dir (MMHOME)>/.wallet	n/a	<Database SID>_<Database schema owner>	<rms schema owner>	Compile, execution	Installer	n/a	Alias hard-coded by installer
RMS forms	DB	<forms install dir>/base/.wallet	n/a	<Database SID>_<Database schema owner>	<rms schema owner>	Compile	Installer	n/a	Alias hard-coded by installer
ARI forms	DB	<forms install dir>/base/.wallet	n/a	<Db_Ari01>	<ari schema owner>	Compile	Manual	ari-alias	
RMWS forms	DB	<forms install dir>/base/.wallet	n/a	<Database SID>_<Database schema owner>	<rwms schema owner>	Compile forms, execute batch	Installer	n/a	Alias hard-coded by installer
RPM app	DB	<RPM batch install dir>/.wallet	n/a	<rms schema owner alias>	<rms schema owner>	Execute batch	Manual	rms-alias	
RWMS auto-login	JAVA	<forms install dir>/base/.javawallet							
			<RWMS Installation name>	<RWMS database user alias>	<RWMS schema owner>	RWMS forms app to avoid dblogin screen	Installer	rwms13inst	
			<RWMS Installation name>	BI_ALIAS	<BI Publisher administrative user>	RWMS forms app to connect to BI Publisher	Installer	n/a	Alias hard-coded by installer

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
AIP app	JAVA	<weblogic domain home>/retail/<deployed aip app name>/config							Each alias must be unique
			aip13	<AIP weblogic user alias>	<AIP weblogic user name>	App use	Installer	aip-weblogic-alias	
			aip13	<AIP database schema user alias>	<AIP database schema user name>	App use	Installer	aip01user-alias	
			aip13	<rib-aip weblogic user alias>	<rib-aip weblogic user name>	App use	Installer	rib-aip-weblogic-alias	
RPM app	JAVA	<weblogic domain home>/retail/<deployed rpm app name>/config							Each alias must be unique
			rpm13	<rpm weblogic user alias>	<rpm weblogic user name>	App use	Installer	rpm-weblogic-alias	
			rpm13	<rms shema user alias>	<rms shema user name>	App, batch use	Installer	rms01user-alias	
			rpm13	<rpm application user one alias>	<rpm application user one name>	App use	Installer	user1-alias	

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
			rpm13	<rpm application user two alias>	<rpm application user two name>	App use	Installer	user2-alias	
			rpm13	<rpm batch user alias>	<rpm batch user name>	App, batch use	Installer	rpmbatch-alias	
			rpm13	<rib-rpm weblogic user alias>	<rib-rpm weblogic user name>	App use	Installer	rib-rpm-weblogic-alias	
ReIM app	JAVA	<weblogic domain home>/retail/<deployed reim app name>/config							Each alias must be unique
			<installed app name>	<reim weblogic user alias>	<reim weblogic user name>	App use	Installer	weblogic-alias	
			<installed app name>	<rms shema user alias>	<rms shema user name>	App, batch use	Installer	rms01user-alias	
			<installed app name>	<reim webservice validation user alias>	<reim webservice validation user name>	App use	Installer	reimwebservice-alias	
			<installed app name>	<reim batch user alias>	<reim batch user name>	App, batch use	Installer	reimbatch-alias	
Alloc app	JAVA	<weblogic domain home>/retail/<deployed alloc app name>/config							Each alias must be unique

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
			<installed app name>	<alloc weblogic user alias>	<alloc weblogic user name>	App use	Installer	weblogic-alias	
			<installed app name>	<rms shema user alias>	<rms shema user name>	App use	Installer	rms01user-alias	
			<installed app name>	<rsl for rms weblogic user alias>	<rsl for rms weblogic user name>	App use	Installer	rsl-rms-weblogic-alias	
RSL app	JAVA	<RSL INSTALL DIR>/rsl-rms/security/config							Each alias must be unique
			rsl-rsm	<rsl weblogic user alias>	<rsl weblogic user name>	App use	Installer	weblogic-alias	
			rsl-rsm	<rms shema user alias>	<rms shema user name>	App use	Installer	rms01user-alias	
SIM app	JAVA	<weblogic domain home>/sim-client/csm							
			rpm	<rpm weblogic user alias>	<rpm weblogic user name>	App use	Installer	rpm-weblogic-alias	
			rms	<rsl for rms weblogic user alias>	<rsl for rms weblogic user name>	App use	Installer	rsl-rms-weblogic-alias	
			rib-sim	<rib-sim weblogic user alias>	<rib-sim weblogic user name>	App use	Installer	rib-sim-weblogic-alias	

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
RETL	JAVA	<RETL home>/etc/security	n/a	<target application user alias>	<target application db userid>	App use	Manual	retl_java_rms01user	User may vary depending on RETL flow's target application
RETL	DB	<RETL home>/.wallet	n/a	<target application user alias>	<target application db userid>	App use	Manual	<db>_<user>	User may vary depending on RETL flow's target application
RIB	JAVA	<RIBHOME DIR>/deployment-home/conf/security							<app> is one of aip, rfm, rms, rpm, sim, rwms, tafr
JMS			jms<1-5>	<jms user alias> for jms<1-5>	<jms user name> for jms<1-5>	Integration use	Installer	jms-alias	
WebLogic			rib-<app>-app-server-instance	<rib-app weblogic user alias>	<rib-app weblogic user name>	Integration use	Installer	weblogic-alias	
Admin GUI			rib-<app>#web-app-user-alias	<rib-app admin gui user alias>	<rib-app admin gui user name>	Integration use	Installer	admin-gui-alias	
Application			rib-<app>#user-alias	<app weblogic user alias>	<app weblogic user name>	Integration use	Installer	app-user-alias	Valid only for aip, rpm, sim
DB			rib-<app>#app-db-user-alias	<rib-app database schema user alias>	<rib-app database schema user name>	Integration use	Installer	db-user-alias	Valid only for rfm, rms, rwms, tafr

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
Error Hospital			rib-<app>#hosp-user-alias	<rib-app error hospital database schema user alias>	<rib-app error hospital database schema user name>	Integration use	Installer	hosp-user-alias	

J

Appendix: Installation Order

This section provides a guideline as to the order in which the Oracle Retail applications should be installed. If a retailer has chosen to use some, but not all, of the applications the order is still valid less the applications not being installed.

> **Note:** The installation order is not meant to imply integration between products.

Enterprise Installation Order

1. Oracle Retail Merchandising System (RMS), Oracle Retail Trade Management (RTM), Oracle Retail Sales Audit (ReSA), Optional: Oracle Retail Fiscal Management (ORFM)

   > **Note:** ORFM is an optional application for RMS if you are implementing Brazil localization.

2. Oracle Retail Service Layer (RSL)
3. Oracle Retail Extract, Transform, Load (RETL)
4. Oracle Retail Active Retail Intelligence (ARI)
5. Oracle Retail Warehouse Management System (RWMS)
6. Oracle Retail Allocation
7. Oracle Retail Invoice Matching (ReIM)
8. Oracle Retail Price Management (RPM)

   > **Note:** During installation of RPM, you are asked for the RIBforRPM provider URL. Since RIB is installed after RPM, make a note of the URL you enter. If you need to change the RIBforRPM provider URL after you install RIB, you can do so by editing the remote_service_locator_info_ribserver.xml file.

9. Oracle Retail Central Office (ORCO)
10. Oracle Retail Returns Management (ORRM)
11. Oracle Retail Back Office (ORBO) or Back Office with Labels and Tags (ORLAT)
12. Oracle Retail Store Inventory Management (SIM)

    > **Note:** During installation of SIM, you are asked for the RIB provider URL. Since RIB is installed after SIM, make a note of the URL you enter. If you need to change the RIB provider URL after you install RIB, you can do so by editing the remote_service_locator_info_ribserver.xml file.

13. Oracle Retail Predictive Application Server (RPAS)
14. Oracle Retail Demand Forecasting (RDF)
15. Oracle Retail Category Management (CM)
16. Oracle Retail Replenishment Optimization (RO)

17. Oracle Retail Analytic Parameter Calculator Replenishment Optimization (APC RO)
18. Oracle Retail Regular Price Optimization (RPO)
19. Oracle Retail Merchandise Financial Planning (MFP)
20. Oracle Retail Size Profile Optimization (SPO)
21. Oracle Retail Assortment Planning (AP)
22. Oracle Retail Item Planning (IP)
23. Oracle Retail Item Planning Configured for COE (IP COE)
24. Oracle Retail Advanced Inventory Planning (AIP)
25. Oracle Retail Integration Bus (RIB)
26. Oracle Retail Point-of-Service (ORPOS)
27. Oracle Retail Markdown Optimization (MDO)
28. Oracle Retail Clearance Optimization Engine (COE)
29. Oracle Retail Analytic Parameter Calculator for Markdown Optimization (APC-MDO)
30. Oracle Retail Analytic Parameter Calculator for Regular Price Optimization (APC-RPO)
31. Oracle Retail Promotion Intelligence and Promotion Planning and Optimization (PI-PPO)
32. Oracle Retail Workspace (ORW)